Extracting Information from Fiction

Sharon Givon

Master of Science

Speech and Language Processing

School of Philosophy, Psychology and Language Sciences

University of Edinburgh

2006

Abstract

Information Extraction (IE) based techniques have great potential to enable companies to leverage valuable information embedded in unstructured textual data. Such data could be exploited to help drive sales and to enhance the customer's experience when searching or browsing for products. Extensive research has been performed in the field of IE; however, to date no work has been directly applied to the domain of fiction. The aim of this study is to explore the ability of IE techniques to extract the central characters and their relationships from the full textual content of works of fiction. To begin our investigation, we present a collection of hypotheses outlining our expectations in approaching and resolving these problems. We then outline our data collection process, which resulted in the creation of a Gold Standard containing ordered lists of characters and their relationships for eight classic book texts. For the task of character extraction, we test two rule-based co-reference resolution models, and two ordering techniques. Our best model achieves an average of 100% coverage on the three most important characters and 78.4% across all central characters, compared to a baseline of 73.3% and 57.4% respectively. For the task of relation extraction, we implement rule-based systems to detect the presence and types of relationships between characters. We achieved 73.3% coverage in detecting the top three pairs of characters involved in relationships. The results for inferring relationship types are preliminary. We provide an analysis of relationship mentions in works of fiction and propose a number of approaches for future work.

Acknowledgments

I would like to thank all the wonderful people who have made this project possible and my time in Edinburgh truly a year I will never forget. Academically, first and foremost I would like to thank Maria Milosavljevic who believed in me from day one and gave me the opportunity to work on this wonderful project. I am thankful for your constant support and encouragement, academically as well as personally. I could not have done this without you. Secondly, I would like to thank Mirella Lapata whose knowledge, enthusiasm and sympathy inspired me in so many ways. I would also like to thank Robert Japp, my supervisor at Amazon.com, for his patience and the time and effort he made to always be there, ready to give advice, feedback or to discuss new ideas.

To everyone at the university who provided help, from the labs of Appleton Tower to Buccleuch Place, and especially to Tom Betts, Thomas Herchenroeder and Sebastian Phelps. I was truly amazed by your endless willingness to help. To my good friend Noemie Guthmann who has taken this journey with me, giving me a home away from home. You are family, friend and study partner all rolled into one.

And last but not least, I would like to thank my friends and especially my family back home for providing moral support and for greatly encouraging me throughout this year. I hope I made you proud.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Sharon Givon)

Table of Contents

Chapter 1: Introduction

"What think you of books?" said he, smiling.
"Books -- oh! no. I am sure we never read the same, or not with the same feelings."
"I am sorry you think so; but if that be the case, there can at least be no want of subject. We may compare our different opinions."

(Pride and Prejudice, Ch. 18, a conversation between Lizzy and Darcy at the second ball at Netherfield)

1.1. Our Aims

This dissertation describes a project which aims to extract information from fiction. The research problems to be addressed in this research were:

1. Can we automatically extract the names of all the central characters involved in a work of fiction?
2. Can we order characters by importance?
3. Can we also extract all of their relationships?
4. Can we order relationships by importance?

Our interest in this problem stems from a desire to provide users with an improved browsing and search experience (particularly for companies such as Amazon.com). By this we mean richer search and browse capabilities, enhanced book recommendation methods and possibly the ability to compare books, and present users with automatically generated book summaries. These aims are based on an intuition that innovative types of information extracted from books (unlike conventional information used for these tasks today such as the names of authors or categorization information) will provide us with information that is indeed useful for these purposes.

1.2. Our Methods

Throughout the research, we were trying to break the problems mentioned above into smaller hypotheses such as “are the most important characters in a work of fiction the most frequent person names?” or “is resolving co-references critical in calculating frequencies of occurrence of characters?”. Our attempt to answer these questions involved not only computation based methods but also linguistics based in-depth analysis of the way the mentioning of characters and relationships is structured within a work of fiction.

1.3. Data Collection

While information extraction is a relatively well studied subject, extracting information from the domain of fiction is an untouched area of research. Therefore, there were no existing annotated corpora we could use in this project. This led us to forming our own data collection process. We selected 18 popular classic books from Project Gutenberg’s digital library, and asked participants to provide us with the information required for this project, such as who the main characters in a book are and what their key relationships are. We performed the data collection experiment from March to May 2006; this process is described in detail in Chapter 4.

1.4. Hypothesis Testing

In addition to having to collect our own data, the lack of existing annotated corpora also made it difficult if not impossible for us to use existing machine learning or statistical methods in order to approach the problem of recognizing characters in fiction. Hence, we developed rule-based methods in order to cope with the challenges of:

a. recognizing person names;
b. resolving co-references between different mentions of the same character; and
c. determining the nature of relationships between characters.

In order to generate lists that reflect the order of importance of the items we extract, we made an assumption that important entities (such as characters and relationships) appear more frequently in the text than other entities. We experimented with two frequency-based methods, one involving pure counts of entities, and the other based on a tf-idf score (term frequency – inverse document frequency). Our methodology and algorithms are described in detail in Chapters 5 and 6.

1.5. Thesis Structure

Chapter 2 describes the motivation behind this project as well as the assumptions and hypotheses we have made. We then provide an overview of other work that is relevant to our challenge and methodologies in Chapter 3. As we mentioned earlier, Chapter 4 describes our data collection process and the final development and test sets of books used in this project. Chapter 5 describes our methodologies, while more detailed and technical implementation issues can be found in Chapter 6. Finally, Chapter 7 discusses our results in relation to our hypotheses described in Chapter 2 and Chapter 8 describes our conclusions and ideas for future work.

Additional information is provided in a collection of appendices. Appendix A describes the structure of the website we used for data collection and other aspects of this experiment. Appendix B contains lists of results for various ordering methods of the Gold Standard. The data we collected along with the information we needed in order to determine what the final sets would be can be found in Appendix C. Appendix D contains all the tables of results that we obtained during our experiments.

This research was carried out in collaboration with Amazon.com.

Chapter 2: Motivation

Information Extraction (IE) is a powerful technology with great potential to help companies leverage the most important information in the unstructured data they have about products they offer. In particular, it can be used to discover information within texts that queries and reports cannot reveal. Amazon.com pioneered the use of data mining for on-sell techniques. The idea of using the buying patterns of customers to suggest other items of interest to a different customer has been highly successful and has been adopted by many other companies since it was first introduced by Amazon.com. Moving to the higher and more complex level of Text Mining allows us to deal with unstructured data in addition to structured data. For example, amongst their latest released features, Amazon.com launched Search Inside! ™ which generates search results that include titles based on full texts of books. Search Inside! ™ has been reported by Amazon.com to significantly increase sales. It also received many positive responses from customers.

2.1. The Challenge

This project aims to take the data mining trend to the next level by exploiting more sophisticated techniques for information extraction. In particular, we want to extract information from full book texts that might be beneficial in presenting customers with a richer and more useful search and/or browsing experience. For example, customers might be interested in cookery books of a particular type, or fiction set in a particular period or with a central character of a particular age, profession, or gender. On a higher level, we aim to approach and set a foundation for more complex tasks such as book summarization and book comparison. This research, being the first of its kind, focuses on an attempt to extract central characters and relationships (particularly person-person relationships between central characters) by exploiting the full texts of books. For example, given the book Pride and Prejudice by Jane Austen, we are interested in extracting an ordered list of central characters such as:

1. Elizabeth Bennet
2. Mr. Darcy
3. Jane Bennet
4. …

We are also interested in extracting an ordered list of relationships (based on the extracted central characters) such as:

1. Elizabeth Bennet & Mr. Darcy – Personal relationship
2. Elizabeth Bennet & Jane Bennet – Family (sisters)
3. …

Amongst the many things you can do with such data is the ability to recommend books based on central character information. For example, we could recommend books about the same central character or books that mention the central characters, or recommend books **about** the author(s) of a given book.

2.1.1. Towards Summarization, Comparison and More

Automatic summarization is the creation of a shortened version of a text by a computer program. The output of this procedure still contains the most important points of the original text (e.g. Goldstein et al. (2000)). However, as expected, in the domain of fiction we would like to be able to do this without revealing the end of the story. The ability to use automatic methods to generate book summaries is a feature that online book sellers well appreciate the need of. Today, customers can learn about the content of a book they are interested in mainly by reading the book's reviews or by searching inside the book. With some applications (such as Search Inside!™) users can search a book or look at the front and back covers. However, what customers seem to really want to know, is what is worth reading and what is useful for a particular purpose. In this case, summaries present the needed information in the most efficient manner.

Being able to compare books is a step beyond book recommendation. Conventional methods involve recommending books based on the author of the book, the categories it is classifed under or by the

buying patterns of other customers. As part of our data collection process (see section 4.2) we present readers with an optional question about their recommendation preferences. In particular, we ask them how they prefer books to be recommended to them when they shop online. Many of the responses include suggestions such as recommending books by "theme", "issues they tackle", "period and style of author", "books on the same author or background[1]" and more. The following response very efficiently reflects the spirit of most of our readers' wishes:

> *"I think books should be recommended by the kind of story they tell; for example, if someone likes romantic period pieces, they might like Pride and Prejudice and A Room with a View. If you like fantastical kids stories, you might like Harry Potter, Lion Witch and the Wardrobe, or His Dark Materials. Categories like "Fantasy" and "Science Fiction" are too broad. There are tons of different kinds of stories within these genres. Maybe someone likes stories with American counterculture themes in the 50s like Capote. Who knows."*

Extracting information from fiction, such as the central characters, their relationships, their attributes, and/or the main story events, can enable sellers to generate recommendations based on quite unconventional features. For example, a book can be recommended based on its setting, the ages of its central characters or the number of events of a certain type that take place in the book (such as murder or war). Books can then be compared and contrasted according to these features and customers can be presented with information such as 'book A is "more romantic" than book B but also "more violent"'. Extracting these features can also contribute to improving and fine tuning book classification. As suggested above, categories can be too broad for some purposes. Our extracted information may be useful for classifying books into more specific categories (e.g. Betts (2006)) or a veriety of categories. For example, a book containing many personal and romantic

[1] For example unlike recommendations based on books written by the same author, here, readers suggest recommending books that are written about a particular author of another book.

relationships and events can be classified as ‘sentimental’, ‘erotic’, or ‘romantic’ and be recommended to potential customers accordingly.

2.1.2. Possible Approaches

Existing approaches to document summarization include information extraction (to identify key entities, relationships and events), natural language generation (in order to work with typed textual phrases, in place of semantics, as input), or sentence labeling implementations that extract sentences from the original document to form a summary. The later approach operates by looking for the most significant information in sentences and labeling them by importance. The most important information is calculated using a set of weighted features such as:

- the location of the text within the original document;
- word and phrase frequency;
- key phrases (e.g. “it is important to say that…”); and
- styling (e.g. boldness or size of font).

(Culotta and Sorensen (2004); Müürisep et al. (2005))

However, current work on automatic summarization is performed mostly on texts that are very different from fictional texts (e.g. news stories, letters or articles). Books are longer and fiction, in particular, often contains many dialogues and direct speech acts. Fiction also contains passages describing feelings, thoughts, actions, events and consequences. Books are also written in different styles, for example some books are written in the first person while others are in the third person.

When we look at existing summaries of books classified as fiction[2], and compare their style to the actual books, it is clear that the differences between these are much greater than the differences between summaries of news stories and the actual news stories. In other words, the distance between a book and its summary is much greater than the distance between a news story and its summary. In particular, we find that:

[2] The summaries mentioned here were downloaded from sites like CliffsNotes or Sparknotes which contain information mostly targeted for students.

- Book summaries are significantly shorter. The summaries we examined consisted of 500-600 words and were approximately 1-3% the size of the original book whereas news summaries are an average of 37% of the length of their sources[3].
- The tense used in book summaries is mostly present simple whereas books use a variety of tenses.
- Summaries never use direct speech acts whereas books include many of them (in the books we use for this research, more than 50% of the sentences are written in direct speech style).
- The language style in book summaries is always modern (regardless of the period and setting of the book) whereas in books the language style varies according to the setting or period.
- Names can be modernized in book summaries (e.g. referring to the Darling family in Peter Pan as "the Darlings").
- In book summaries, referring to a certain character is done mainly by using a single name. However, in books, characters can be referred to by many different names.

Dealing with such different types of texts and the need to create summaries that are completely different in nature to their source introduces a large range of complexities. These complexities cannot be approached using current summarization techniques whether by sentence extraction (merely copying the information deemed most important by the system to the summary) or abstraction (regeneration of the extracted content). Abstractive approaches, on the other hand, use information extraction, ontological information, information fusion, and compression. When working with specific domains, a set of "interests" can be defined, for example in the 'terrorist attack' domain we can define interests such as time, place, number of casualties etc. The system then attempts to find this information in the original text and includes it in the summary (DeJong (1978); Rau and Jacobs (1991)). Abstractive methods seem more appropriate to our task than extraction based methods and may also prove to

[3] Taken from DUC (Document Understanding Conferences) 2005 http://www-nlpir.nist.gov/projects/duc/duc2005/

be useful for the domain of fiction. For example, text written in direct speech style cannot be inserted directly to the summary. It has to be at least converted into the indirect speech form in order to match the style of book summaries. However, as already shown above, dealing with books requires more sophisticated natural language analysis to identify key passages. An analysis of word relatedness or discourse structure may prove useful here.

Although abstraction based summarization methods seem like a natural choice for book summarization, true abstraction involves taking the process one step further to recognize that a set of extracted passages together constitute something new that is not explicitly mentioned in the source, and then replacing them in the summary with the new concept. Since the new material is not mentioned explicitly in the original text, the system must have access to external information of some kind, such as an ontology or a knowledge base (these are referred to as rich-knowledge systems). But since no large-scale resources of this kind yet exist, abstractive summarization has not progressed beyond the proof-of-concept stage.

Identifying the key elements in a book is crucial to the process of automatic summarization. In order to generate an effective summary, we need to identify information such as the main characters, the relationships between them and how these relationships develop over time, as well as main story events. Therefore, we believe that the outcome of this research can serve as a platform for the development of tools that can be used for automatic summarization and comparison of books.

2.2. Our Hypotheses

In order to achieve our goals, we begin by listing our key assumptions and hypotheses which we aim to prove or disprove in the work that follows. We have divided our hypotheses into three distinct units of work.

2.2.1. Central Characters Recognition

Our first challenge involves identifying all the characters in a work of fiction. Here, we make our first hypotheses, as listed below.

Hypothesis 1

The importance of characters within a work of fiction is directly proportional to their frequency of mention in that work of fiction.

All of our hypotheses related to character extraction are based on this key hypothesis, which we will aim to test in this work. The hypotheses below differ in their methodologies for detecting characters and co-references of characters, but they are similar in the aspect of computing frequencies in order to obtain a final ordered list. In books written in the first person we still expect this to be true, although in this case Proper Names are replaced with first person personal pronouns like ‘I’, ‘my’, ‘me’, or ‘myself’.

Hypothesis 2

Named Entity recognition (NER) techniques can be used to find a significant number of the mentions of key characters in a work of fiction.

Named entities are phrases that contain the names of persons, organizations, locations, times or quantities. Existing state-of-the-art methods obtain an F-score (see section 7.2.3) of over 88% on English texts although none of these methods are trained or tested on texts that are similar to books. We assume that by applying existing NER methods to book texts we can succeed in extracting person type entities.

Hypothesis 3

The most important characters in a work of fiction are the most frequently occurring person named entities.

After applying NER methods to book texts, and by computing the frequency of each detected named entity, we hypothesize that we can generate an ordered list of central characters where the more important characters appear at the top of the list.

Hypothesis 4

Identifying all the sequences of Proper Nouns in a work of fiction can provide a list of important named entities.

By relying on existing text processing tools, we believe we can successfully detect the most significant sequences of Proper Nouns in a text. These include nouns that refer to particular entities such as specific people, locations, or organizations. However, as will be stated in hypothesis 5, we expect that central characters will appear more frequently in the text, thus allowing us to exclude irrelevant names appearing less frequently.

Hypothesis 5

The most important characters in a work of fiction are the most frequent Proper Nouns.

We expect sequences of Proper Nouns that refer to central characters to appear more frequently in the text than sequences of Proper Nouns referring to less important characters or to other entities such as places or organizations. After computing frequencies for each sequence of Proper Nouns, we expect to obtain an ordered list where central characters appear at the top, above a certain threshold (e.g. within the ten most frequent names).

Hypothesis 6

Normalizing simple character mention frequencies by corpus-wide frequency (tf-idf) will provide a more accurate list of central characters than simple frequencies alone.

By computing a tf-idf score (term frequency – inverse document frequency), for each unique term, we expect to obtain a more accurate list of characters, especially in terms of absolute order. The tf-idf score increases with character importance since it is based on the frequency of the term in relation to the frequency of the rest of the terms in the book, normalized by its uniqueness in relation to terms in other books in the corpus. We expect that ordering the terms

by their tf-idf score will result in an ordered list where the central characters appear at the top of the list.

2.2.2. Co-reference Resolution

We now turn our attention to the problem of co-reference resolution. In particular, we need to identify all the important mentions of a character in order to obtain a true measure of its frequency within a book. Our first hypothesis here is based on the assumption that the problem of central character identification cannot be resolved accurately without first solving this issue.

Hypothesis 7

Calculating the frequencies of occurrences of character mentions within a work of fiction cannot be performed accurately without resolving co-referential mentions of person names.

We treat the task of co-reference resolution as a pre-requisite in extracting an ordered and distinct list of central characters. In long texts, such as books, entities are often referred to by more than one name. In order to compute frequency or frequency-like scores for each character, we have to find all the co-references and unify all those referring to the same character. We will attempt to resolve the co-references of person names by using a rule-based approach to identify the different parts of the names.

We now make two separate hypotheses regarding how we might resolve ambiguous character mentions. We expect that only one of these will be true, so we number them 8A and 8B.

Hypothesis 8A

Person names appearing in a work of fiction that may refer to more than one character are more likely to refer to a more frequent character in the book rather than to the less frequent character.

For example, in a book involving more than one family member (e.g. Mr. Darcy and Miss Darcy) we assume that in cases where an

ambiguous version of the name is used (e.g. just ‘Darcy’), it refers to the more frequent name in the book (in the case of Pride and Prejudice, to Mr. Darcy rather than to Miss Darcy, his younger sister). By preferring characters with higher term frequency (tf) values (according to their current frequency count) we expect to obtain a more accurate list of characters, mainly in terms of absolute order.

Hypothesis 8B

Person names appearing in a work of fiction that may refer to more than one character are more likely to refer to the character appearing closer to them in the book rather than to the character which is less close.

Given an ambiguous character mention, we expect that the name will refer to the character mention appearing closest to it in the text.

2.2.3. Relationships

The final problem that we will approach is that of identifying the most important relationships between our central characters. Again, we will adopt a frequency-based approach, as evidenced in our first hypothesis.

Hypothesis 9

The importance of relationships between central characters in a work of fiction is directly proportional to their frequency of mention.

We expect mentions of important relationships in the text to be more frequent than less important relationships.

Hypothesis 10

Important relationships and the nature of the relationship between characters in a work of fiction can be identified within the context of single sentences, and in particular sentences that are written in direct speech style and contain at least one descriptive word between character mentions.

Existing research in Relationship Extraction (RE) focuses mainly on single sentences (Culotta and Sorensen (2004)). In particular, it assumes that important information can be found in single sentences in a text. We therefore make this assumption and in addition require the sentences to be written in an indirect speech form. About half of the sentences in books categorized as fiction can be defined as written in direct speech (a dialogue style). The use of direct speech must have a major impact on any attempt to extract relationships. In particular, it will requires more complex analysis, such as detecting who the speaker and receivers are in a speech act, who/what they are talking about, or who else is witnessing the dialogue. For the scope of this research, we will focus on indirect speech sentences only.

Hypothesis 11

The importance of a relationship between two characters is directly proportional to the frequency of occurrence of sentences which mention the two characters in a work of fiction, regardless of the content of those sentences.

By computing frequencies of sentences that contain exactly two central characters we expect to infer that the most frequent pairs of characters have a relationship of some kind, without identifying the type of relationship.

Hypothesis 12

Analyzing the descriptive words appearing between two character mentions in individual indirect speech sentences will provide sufficient information for indicating what type of relationship exists between those characters.

By descriptive words we refer to words tagged as nouns or verbs, based on the part of speech tags assigned to them at an earlier stage.

Hypothesis 13

By computing an average of the similarity distance between the content words found in the context of two character mentions and the content words used to describe our main relationship types, we

can determine the most likely relationship type between those two characters.

Once we detect sentences that we believe define relationships, we will attempt to classify the relationship type using existing off-the-shelf tools to compute the similarity distance between descriptive words in the sentences and the words describing the relationship types. We expect that from a computed average of the similarity distance between descriptive words and relationship type, we can infer the most likely relationship type.

Hypothesis 14

We can define a set of keywords which will be an indication of particular relationships, and use these to accurately predict relationship types if these keywords appear in the context of two characters in a work of fiction.

The relationship type between two characters may be more accurately determined by using keyword-based concept rules. If we fail to use a similarity distance measure, we will attempt to use stricter hand-crafted rules that define concepts by relating specific descriptive words found in the sentence to the relationship types.

Chapter 3: Background and Related Work

Over the last decade, extensive research has been performed in the field of information extraction (IE). Much of this research has focused on named entities, primarily within newswire articles and bio-medical data. Relationship extraction is a relatively new and as yet unsolved area of research within IE. No research in the area of information extraction has been directly related to the domains of fiction or literature, or to texts of such length and nature. This project therefore challenges the boundaries of IE technologies by addressing the under-researched problem of information extraction from fiction. We expect it to further pioneer solutions to using full length book texts.

3.1. Named Entity Recognition (NER)

Named entities are phrases that contain the names of persons, organizations, locations, times and quantities. The example in Figure 1 contains entities of three types: ORG (organization), PER (person) and LOC (location).

[ORG U.N] official [PER Ekeus] heads for [LOC Baghdad]

Figure 1: NER example

Named entity recognition (NER) is a subtask of information extraction that seeks to locate these phrases in text and classify them into some predefined categories (such as the names of persons, organizations, locations, expressions of time, quantities, monetary values, percentages, and more). There are two main approaches to NER, the rule-based approach and the machine learning approach.

3.1.1. Machine Learning Methods

Machine learning methods to NER are implemented using probabilistic matching and machine learning models. They are trainable with training

files. For example, a NER system can consist of Hidden Markov Model (HMM)-based models. Here, within each state (nameclass) a bigram language model is used for computing the likelihood of words occurring within that nameclass and the likelihood of every word is based solely on the previous word ("Markov chain") (Zhou (2002)).

Many successful methods for NER use an implementation of Maximum Entropy. The maximum entropy framework estimates probabilities based on the principle of making as few assumptions as possible, other than the constraints imposed. Such constraints are derived from training data, expressing some relationship between features and outcome. The probability distribution that satisfies the above property is the one with the highest entropy. It is unique, agrees with the maximum-likelihood distribution, and has the exponential form (Della Pietra et al. (1997)).

3.1.2. Rule-Based Methods

Rule-based NER methods consist of definitions of patterns and actions (for entity tagging). Rule-based NER systems use rules that are handcrafted by domain experts. The rules use internal and external context and can be implemented using regular expressions or similar mechanisms. The rules must be maintained continuously. Among rule-based NER systems we can find GATE (Cunningham et al. (2002)), and SProUT (Becker et al. (2002)). Table 1 shows two rule examples where Xxxx+ is a sequence of capitalized words, JJ* is a sequence of one or more adjectives, and PROF is a profession.

Context Rule	***Assign Label***	***Example***
Xxxx+ is a? JJ* PROF	PER	Yuri Gromov, a former director
Xxxx+	PER	White himself

Table 1: Example for NER rule

3.1.3. Comparison

Rule-based methods consist of simpler algorithms and in this respect they are more effective. They also outperform machine learning based methods

by around 2%. However, in terms of the work required, rule-based systems require linguistic or domain experts in order to form the handcrafted rules while machine learning based methods rely mainly on the existence of training data. While machine learning based algorithms can be adopted quite easily to other domains or languages (by retraining on some annotated training data), in rule-based systems, the rules have to be modified or re-written in order to be adopted to new data types. To conclude, in terms of labor, machine learning based systems might be cheaper (since human labor is expensive), however they will require specific training data, and this is also very time consuming to generate.

3.1.4. Hybrid Methods

In CoNLL 2003[4] the shared task[5] included NER in both English and German where the English data consisted of a collection of newswire articles from the Reuters Corpus. The best results obtained an overall F-score of 88.76% (and in particular 93.85% on PER entity type). The winning solution used features such as part-off-speech (POS) tags, lexical features, affix information (n-grams), previously predicted NER tags, orthographic information, gazetteers, chunk tags, and global case information and consisted of a combination of four classifiers (Florian et al. (2003)).

3.1.5. Discussion

From the shared task of CoNLL 2003 we can see that NER has reached a mature stage. In our research we aim to explore how efficient these extraction methods are when implemented on fiction book data in general, and how we can infer a list of central characters from the extracted named entities in particular. However, one of the biggest problems with machine learning based NER methods is the fact that they are domain specific. They

[4] The CoNLL website: http://www.cnts.ua.ac.be/conll2003/proceedings.html

[5] A shared task is a defined task performed by participants from different backgrounds and/or different skill levels. The shared task is a competition like procedure where participants aim to find methods to achieve the best results given the provided data sets.

are trained on specific domains and are targeted to label texts from these domains. For example, they can be trained on a newswire corpus or on biomedical texts and perform relatively well on texts from such domains. This may present us with problems if we want to use an off-the-shelf NER system. Statistical based systems can be trained on new domains, but we lack an existing annotated corpus that we can use to train such a system on. For the kind of data we deal with in this project, and given our constraints, a solution that consists of rule-based NER methods might be more adaptive. However, writing a good set of rules that covers the majority of person named entities in a work of fiction is a time-consuming task on its own.

We are thus constrained to use existing methods that are trained on different domains and a large variety of entity types than the ones we are interested in, which may significantly affect our performance.

3.2. Co-reference Resolution

Information Extraction (IE) systems are designed to extract fixed types of information from documents in a specific language and domain. Co-reference resolution aims to identify equivalence between the named entities that appear in a text. As a result, all references to the same entity are grouped into a co-reference chain. Most of the work done on co-reference deals with a single language and a single text document (usually newswire).

The co-reference problem can be treated as two separate challenges: orthographic co-reference between named entities or names in the text, and pronominal co-reference resolution which deals with the occurrence of pronouns. Resolving orthographic co-reference between names in the text is mainly done using a set of handcrafted rules. These rules can be generic and suit all kinds of entities found in the text, or they can be entity type specific (e.g. relevant to person names only). Results reported for orthographic based methods vary at around 96% of precision and 93% of recall (see, e.g. Bontcheva et al. (2002)).

The task of pronoun resolution is much more difficult than orthographic resolution. Approaches to pronominal resolution which are defined as “knowledge poor” intend to provide inexpensive (in terms of the

cost of development) and fast implementations that do not rely on complex linguistic knowledge, yet they work with a sufficient success for practical tasks (e.g. Mitkov (1998)). These methods consist of steps such as identification of the context of the pronoun, inspecting the context for candidate antecedents that satisfy a set of consistency restrictions, assigning salience values to each antecedent based on a set of rules and factors and finally choosing the candidate with the best salience value. The implementation mainly relies on the part-of-speech information, named entity recognition and orthographic co-reference information, or in some cases, on syntax parsing, focus identification or world-knowledge based approaches (such as Humphreys et al. (1998) and Lappin and Leass (1994)). The results reported for the pronominal co-reference resolution are significantly lower at around 66% precision and 49% recall (Mitkov (1998); Bontcheva et al. (2002); Barbu and Mitkov (2001)).

Given the relatively low results obtained by pronoun resolution solutions and the time restriction of this project, we believe that implementing a rule-based method for orthographic co-reference is realistic and useful in the domain of fiction. In particular, we are interested mainly in person type entities, for which we can obtain access to external lists and gazetteers. Therefore, building a set of rules for co-reference resolution of person names is a task that can be implemented within the scope of this project.

3.3. Relation Extraction

The purpose of relation extraction is to recognize relationships between entities in unstructured text. As described earlier, an entity is commonly a physical thing such as a person, place, organization, gene or chemical name, but can also be more abstract and encompass things such as prices, units, measurements, schedules and even philosophical beliefs. Here we describe three main methods for relation extraction.

3.3.1. Semi-Supervised Machine Learning

Agichtein et al. (2000) presented Snowball – a system for extracting relations from unstructured text. Snowball introduces novel strategies for generating patterns and extracting binary relationships from plain-text documents. This technique requires only a handful of training examples from users. These examples are used to generate extraction patterns that, in turn, result in new relationships being extracted from the document collection. At each iteration of the extraction process, Snowball evaluates the quality of the patterns and relationships without human intervention, and keeps only the most reliable ones for the next iteration. The primary advantage of Snowball, and other related semi-supervised training systems, is that they require little to no human annotation. Nevertheless, the relationship type that Snowball focuses on is the [organization, location] (e.g. [Microsoft Headquarters, Seattle]). Furthermore, Snowball relies on an intrinsic property of organizations and locations (that every organization has its headquarters in only one location) when calculating the confidence score of a pattern. However, this property does not hold for all relations, let alone the relationship types that we are interested in that may be extremely less structured than organization-location type relations. Furthermore, a character in a book can be involved in more than one relationship of the same type with more than one other character and this contradicts the intrinsic property assumption of Snowball.

3.3.2. Integrated Parsing

Miller et al. (2000) present a model which encodes all decisions into natural language parse trees. Parse trees can be broken down into the root node (usually S – the sentence), internal nodes, pre-terminal nodes and terminal nodes. The terminal nodes (the leaves of the tree) are the words of the sentence. The pre-terminal nodes represent the part-of-speech tags for each word, such as noun, preposition, determiner etc. Once this is done, the model of interest becomes P(T,S), where T is a parse tree and S the input sentence. In particular, they used lexicalized parse trees where a head word (the most representative word of the phrase that an internal node subsumes)

is assigned to each internal node of the tree. For the purpose of parsing, a Collins parser model is used (Collins (1997)) although their implemented model is different from the original one in many significant ways (e.g. generating word features for unknown words, such as capitalization, as a way to improve entity recognition). In order to create a large set of labeled examples, Miller et al. ran the Collins parser, trained on the Penn Treebank (Marcus et al. (1993)), over their training data. The trees are then annotated with entity and relation information. Finally, it is possible to retrain the Collins parser on the augmented trees in order to tag new sentences.

The intuition behind the integrated parsing approach seems sound. Every entity, relation, part of speech, and parse tree decision is related and they should all be made at the same time. In particular, information about relations, entities and tree structure are highly correlated. A clear disadvantage of this model is that it relies heavily on a large set of manually annotated examples (which currently do not exist for our research). Moreover, the model used by Miller et al. is also a sentence-based model and on the surface appears to only manage simple relations. Complex relations can exist over multiple sentences making such relations difficult to extract from a sentence-based system.

3.3.3. Kernel Methods

Zalenko et al. (2003) present a model for relation encoding that is based on shallow parsing. Shallow parsing (or "light parsing") is an analysis of a sentence which identifies the constituents (noun groups, verbs, etc.), but does not specify their internal structure, nor their role in the main sentence. Given shallow parsed sentences, it is possible to create a set of positive and negative examples for classification. Having extracted these examples it is fairly straightforward to create a classifier to identify sub-trees containing the relation of interest (e.g. by using a kernel similarity function over input pairs). Unlike the integrated parsing method, this method enables the incorporation of long-range features into relation detection decisions, representing a rich set of dependent features in a computationally tractable manner. Another advantage is that by reformulating the problem into a

yes/no classification problem, they are able to take advantage of state-of-the-art discriminative methods such as support vector machines. However, like the model used by Miller et al., this requires large sets of annotated training data which we cannot provide at this stage.

3.3.4. Discussion

This review does not mention the empirical performance of each system on their varying tasks. This is due to the fact that each system is evaluated with different data and relations of interest. Comparing numbers would be, at best, misleading. Here we focus on the qualitative aspects of each system. However, it would be an interesting experiment to evaluate each system on the same set of data to check empirical performance. Performance numbers can also be retrieved from so-called 'bake-off' competitions that serve just this purpose, such as MUC[6] (a series of Message Understanding Conferences) and its new incarnation ACE[7] (Automatic Content Extraction).

We will attempt to use sentence-based methods in our work. However, none will be ideal given the scope of our project, the nature of the available data, and time limitations. For example, we fully expect that the parsing task will be particularly difficult. The language used in fiction (and particularly in the books we work with which were written in the 18th and 19th centuries) is very different from other language such as the one used in news stories or articles. At this point we do not expect parsers trained on modern data to perform well on texts that humans may find difficult to read. Instead, we aim to investigate the behavior of relationships and their patterns in the domain of fiction in order to shed some light on what the main challenges and obstacles are and lay ground for further research and solutions.

[6] http://cs.nyu.edu/cs/faculty/grishman/muc6.html

[7] http://www.nist.gov/speech/tests/ace/index.htm

Chapter 4: Data Collection and Pre-Processing

4.1. Project Gutenberg

Our project needs to be developed and tested on full book texts. For this purpose, we use the free texts available on Project Gutenberg. Project Gutenberg[8] (often abbreviated as PG) is a volunteer effort to digitize, archive, and distribute cultural works. Founded in 1971, it is the oldest digital library, and most of its items are the full texts of public domain books. PG attempts to provide free items in its collection, in long-lasting open formats that can be used on almost any computer. Books in PG are classified in the Library of Congress classification system. We focus on books under the following Library of Congress Classes (LoCC):

- PS: Language and Literatures: English literature; and
- PR: Language and Literatures: American literature.

We use 18 books, downloaded from PG for our data collection. Our final development set includes three books and the test set includes five books (see section 4.2). We also use the texts of a thousand books, downloaded from the American Literature category, in order to compute the idf value in the tf-idf score (see section 5.2).

4.2. Data Collection

We mentioned earlier that there is a lack of former IE research in the domain of fiction. One consequence of this is the fact that there is no annotated corpus available in the domain of fiction. Therefore, we have to establish a Gold Standard set for developing and testing our system. A Gold Standard, by definition, is a set of texts where instances are annotated with their related ontological concepts. Our first step therefore, is to collect relevant data for each book we use in this research. This data consists of

[8] http://www.gutenberg.org/

ordered lists of central characters and relationships between central characters. For the purpose of this research we only use the collected lists. However we see the creation of an annotated corpus as a primary task for future research.

A set of 18 classic titles were used to collect data. Since we preferred books that most people were familiar with, we selected the titles according to their sale records on Amazon.com. To collect the data we used a website[9] where users could choose a title from a given list and fill in a form with the required information.

4.2.1. Collecting the Data: First Attempt

We performed our data collection in two phases. Our initial design consisted of the following:

Books

We had the following 12 titles available to choose from on the website:

- A Tale of Two Cities (by Charles Dickens)
- Alice's Adventures in Wonderland (by Lewis Carroll)
- Anna Karenina (by Leo Tolstoy)
- Bleak House (by Charles Dickens)
- Crime and Punishment (by Fyodor Dostoyevsky)
- Emma (by Jane Austen)
- Great Expectations (by Charles Dickens)
- Oliver Twist (by Charles Dickens)
- Pride and Prejudice (by Jane Austen)
- The Adventures of Huckleberry Finn (by Mark Twain)
- The Count Of Monte Cristo (by Alexandre Dumas)
- The Wonderful Wizard of Oz (by L. Frank Baum)

Main Characters

In this section we had seven open slots for users to type in names of central characters. Users were guided to order the characters according to their view of their importance to the story

[9] http://sgivon.tripod.com/Index.html

Relationships

To help users to define relationships we used relationship definitions from ACE[10]. ACE (Automatic Content Extraction) is a program whose objective is to develop automatic content extraction technology to support the automatic processing of textual data. The ACE program, carried out by the Speech Group of NIST (National Institute of Standards and Technology), is dedicated to the development of technologies that automatically infer meaning from language data. Relation recognition is one of the tasks in ACE, and its definition[11] states seven relation types and their sub-types to be extracted as shown in Table 2.

Type	***Sub-Type***
ART (artifact)	User-Owner-Inventor-Manufacturer
GEN-AFF (Gen-Affiliation)	Citizen-Resident-Religion-Ethnicity, Org-Location
METONOMY	None
ORG-AFF (Organization-Affiliation)	Employment, Founder, Ownership, Student-Alum, Sports-Affiliation, Investor-Shareholder, Membership
PART-WHOLE (part-whole)	Artifact, Geographical, Subsidiary
PER-SOC (person-social)	Business, Family, Lasting-Personal
PHYS (physical)	Located, Near

Table 2: ACE relationship types

In our research we focus on the main characters in books and primarily in relationships between the main characters. Furthermore, we narrow down the scope of the task to relationships between exactly two central characters and to the sub-type relationships of ACE's PER-SOC type:

1. Family
2. Personal

[10] See http://www.nist.gov/speech/tests/ace/index.htm

[11] The relation types are takes from ACE05 (ACE 2005 Evaluation) plan: http://www.nist.gov/speech/tests/ace/ace05/doc/ace05-evalplan.v3.pdf

3. Business
4. Other

The above four types cover the important human relationships between two characters. We used these types in the Relationships section on the website form. We allowed users to select pairs of characters (from two drop down lists with numbers from 1 to 7), and then the relationship type, again from a drop-down list, offering the four options mentioned above. Selecting the option 'Other' activated a text box where users could describe the relation.

Events[12]

This section was defined as optional. Here users were given four slots consisting of text-boxes where they were asked to describe events using their own words. Users were guided to order the events according their view of their importance to the story (rather than chronologically).

For each book we provided a link to the full book text, downloaded from Project Gutenberg.

When analyzing the responses we collected, we identified the following problems:

Books with no responses

For some of the books on our list we received no responses at all.

"Too many characters"

It seemed that many users were forcing themselves to fill in all the available slots in the main characters section. We originally allowed up to seven characters under the assumption that readers would tend to state on average three or four characters. However, the majority of the responses included definitions of seven main characters. We treated this phenomenon as problematic since one of the questions that this project attempts to answer is how a central character is defined and one way to explore this is by learning what readers consider to be a main character.

[12] Event Extraction was defined as an optional task for the scope of this project. Due to time restriction we did not use the information collected for events but we intend to use it in future research as described in Chapter 8.

Only by giving readers free hand to some extent when choosing their main characters could we truly learn from their responses.

Vague relationship information

The information readers gave in the relationship section was too vague. Detailed information was given only when readers selected the option 'Other'. We realized that for the further analysis that was required for relation extraction we needed a deeper level of information for each relationship.

Events described in multiple and inefficient formats

Although the events section was defined as optional, many readers opted to fill it in. However, the responses in this section varied from very short and general (e.g. *"the courtship between Elizabeth and Mr. Darcy"*) to very long texts with superfluous information or multiple events joined into one (e.g. *"Darcy falls in love with Lizzie and proposes. She refuses him. Mr. Bingley moves to Netherfield and is joined by his friend Mr. Darcy and his sisters and brother in law"*).

Considering the responses we received, and in order to resolve the problems described above, we altered the design of the form used on the data collection website.

4.2.2. Collecting the Data: Second Attempt

The following changes were incorporated into the data collection website[13]:

Books

The popular books from the first phase were kept, and books that received no responses were removed. Instead, new titles were added. The following 13 titles were available:

- A Tale of Two Cities (by Charles Dickens)
- Anna Karenina (by Leo Tolstoy)
- Crime and Punishment (by Fyodor Dostoyevsky)
- Emma (by Jane Austen)

[13] This is the design used on the website as of August 25, 2006.

- Great Expectations (by Charles Dickens)
- Jane Eyre (by Charlotte Bronte)
- Little Women (by Louisa May Alcott)
- Oliver Twist (by Charles Dickens)
- Peter Pan (by J. M. Barrie)
- Pride and Prejudice (by Jane Austen)
- Treasure Island (by Robert Louis Stevenson)
- Uncle Tom's Cabin (by Harriet Beecher Stowe)
- Wuthering Heights (by Emily Bronte)

Main Characters

This section was divided into two parts. The first part was defined as mandatory[14] where readers were asked to fill in names of three main characters. The separation was visual both in color and by a separating line. Another four optional slots were kept for more characters.

Relationships

The text box for description was activated by default and readers were asked to add a description by using this box for every relation they add. An example was provided.

Events

The original design was replaced by a more user-friendly interface that attempted to encourage the user to describe events in an efficient manner. We adopted the same mechanism we used for the relationship definition and turn to ACE 2005's event recognition task definition for event types. ACE defined the events types and sub types shown in Table 3.

[14] Although this was defined as mandatory, no validation rules were applied to it to force the user to fill it in.

Type	*Sub-Type*
Life	Be-Born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-Ownership, Transfer-Money
Business	Start-Org, Merge-Org, Declare-Bankruptcy, End-Org
Conflict	Attack, Demonstrate
Contact	Meet, Phone-Write
Personnel	Start-Position, End-Position, Nominate, Elect
Justice	Arrest-Jail, Release-Parole, Trial-Hearing, Charge-Indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

Table 3: ACE event types

Using the same methodology implemented in the relationships section, in this section we focused on events that could be detected between main characters only (two human characters) and particularly on life events. We provided readers with five slots for events. For each event users first had to choose an event type from a drop-down list. The list contained event types which served as categories (readers could not choose them) and sub types which readers could choose from:

1. Life
    a. Be born
    b. Divorce
    c. Injure
    d. Die
2. Conflict
    a. Attack
3. Contact
    a. Meet
    b. Phone
    c. Write
4. Other
    a. Other

In total, we allowed choosing from nine event types. Once chosen an event type, users were asked to specify one or two central characters involved in the event. Like in the relationships section, this too was done by selecting from two drop-down lists containing numbers from 1 to 7, corresponding to the central characters section. Finally users could add event description in the provided text-box.

4.3. Final Test Data

From the original list of 18 titles, we collected user responses for 15 titles. Of these 15 titles, only eight titles had a significant amount of data and they serve as our Gold Standard. We use three titles as a development set and the remaining five titles make up the set that is used for testing. (See Appendix C)

Development Set:

1. Pride and Prejudice
2. Peter Pan
3. Jane Eyre

Test Set:

1. Wuthering Heights
2. Alice in Wonderland
3. Little Women
4. Great Expectations
5. Emma

4.4. Agreement Tests

4.4.1. Central Characters

In order to measure the agreement between different users, we compute agreement on the absolute order of central characters. Prior to computing agreement however, we exclude statistically irrelevant data according to the following criteria:

- characters voted for by less than 27%[15] of the total number of readers filling in responses for the book; and
- reader responses consisting of less than three characters (since readers are instructed to fill in at least three central characters).

Agreement is then computed between all possible pairs of readers, giving the results described in Table 4.

For absolute agreement we count the number of times when readers gave a character the same rank. This is divided by the maximum number of characters ranked by the readers (thus excluding characters that are not ranked by the pair of readers).

Book	***Top 1***	***Top 2***	***Top 3***	***Superset***
Wuthering Heights	100%	100%	66.6%	58.3%
Pride and Prejudice	82.9%	80.2%	66.6%	58.1%
Peter Pan	100%	75%	66.6%	51.1%
Jane Eyre	100%	100%	70%	78.9%
Emma	100%	100%	73.3%	50.4%
Great Expectations	100%	83.3%	71.1%	46.1%
Little Women	100%	65%	50%	36.6%
Alice in Wonderland	100%	100%	0.833	63.3%
Average	**97.8%**	**87.9%**	**68.4%**	**55.3%**

Table 4: Absolute order agreement on central characters

A good agreement is considered to be above 60% since 50% represents chance, and in agreement studies 60% is considered adequate. Therefore, from the average figures shown, we can conclude that readers highly agree on the top three main characters in books and that the agreement is stronger on the top two characters and top character in the book. However, there is no significant agreement on the whole superset of voted characters.

[15] Our goal was to use a single value as a threshold which would allow us to keep popular characters (and also relationships) but exclude ones that received only a few votes. We experimented with a range of values and finally used human judgment to make a decision on which single value can be used to obtain good lists for all the books.

4.4.2. Relationships

In order to measure the relationship agreement level between users, we again compute agreement on absolute order. We exclude the same readers excluded when computing agreement for central characters and use the same conditions applied to central characters to remove relationships that are voted for by less than 27%[15] of the total number of readers. Since in the relationships section we had no strict requirement to fill in a minimum number of relationships, any responses containing between one to the maximum of five relationships are included.

For absolute agreement we count for each pair of readers, the number of times they selected the same relationship type between the same two characters at the same rank. This is divided by the maximum number of relationships defined by the two readers (thus excluding characters that are not ranked by the pair). The results are shown in Table 5.

Book	***Top 1***	***Top 2***	***Top 3***	***Superset***
Wuthering Heights	50%	33.3%	22.2%	23.6%
Pride and Prejudice	63.6%	44.7%	33.7%	66.6%
Peter Pan	0%	0%	0%	15.2%
Jane Eyre	40%	30%	20%	49%
Emma	66.6%	36.6%	31.1%	56%
Great Expectations	26.6%	16.6%	11.1%	26.1%
Little Women	16.6%	16.6%	11.1%	23.6%
Alice in Wonderland	33.3%	33.3%	33.3%	25%
Average	**37.1%**	**26.3%**	**20.3%**	**35.6%**

Table 5: Absolute order agreement on relationships

We mentioned earlier that a good agreement is considered to be above 60%. Therefore, from the above figures we can conclude that there is no significant agreement on relationships between central characters.

The method used for collecting the data may be responsible for the lack of agreement on relationships between central characters, and perhaps also on the central characters. The data collection process is structured in a

way whereby readers can freely type in names of characters and define any relationships they can think of. This may strongly affect their responses. We assume that if the experiment had been designed in a different way where pre-defined lists of characters were available for readers to rank, and possibly pre-defined lists of relationships to rank, the agreement level could have been higher. Additionally, we assume that our results are not as good as they could have been had the experiment been different. We expect this to affect the results in terms of coverage and absolute order but mainly affect the relationship extraction results since relationship definitions are based on central characters. However, for the scope of our project, given the strong agreement on central characters and our objectives regarding relationship extraction, we consider this data sufficient.

4.5. Ordering the Lists

The purpose of our collected data was to serve as a Gold Standard which we can compare our results to. In order to do that, we need ordered sets of characters and relationships that we can compare our ordered results to. The main problem with computing the best order is that in some cases there are gaps in the data, for example in cases where a reader does not vote for one of the characters in the superset. This can happen since readers type in character names and relationships rather than ordering predefined lists. Therefore, readers do not necessarily rank every character or relationship in the superset of characters. We assess two methods for ordering the collected sets for each book.

4.5.1.1. Order by Median

For each character or relationship in the superset of characters we replace missing ranks with a large value (we chose ‘100’). We collect and order all the ranks for each character or relationship and compute the median value. Finally we order the lists by the median values.

4.5.1.2. Order by Average Score and Number of Voters

Using this method we manage to take into consideration both the average score of the characters or relationships based on given ranks as well as to compensate for the value of the number of voters. We want the final rank to be higher as the number of readers ranking the characters or relationship is lower (a higher rank represents lower importance). We compute a final rank for each character and relationship using the following formula:

$$Item\_Rank = Item\_Ave\_Rank * \frac{Total\_Num\_Voters}{Item\_Num\_Votes}$$

where:

- Item_Ave_Rank is the average of all ranks given to the character or relationship;
- Total_Num_Voters is the total number of reader responses for the book; and
- Item_Num_Votes is the number of readers giving ranks to the character or relationship.

4.5.1.3. Final Ordering

Our two ordering methods always agree on the top three items and generally give quite similar results. However, the second method resolves ties among ranks where the median does not. Since we are interested in examining our system's performance on the top three characters and relationships as well as on the whole set we choose to use the order generated by the second method, based on the average score and number of users. (See Appendix B).

4.6. Data Pre-processing

All of our experiments use a set of existing pre-processing methods. The quality of the pre-processing stage is crucial to the results of Information Extraction. Here, we provide a brief review of the tools and methods used for text processing in our research.

The purpose of the pre-processing stage is to take as input raw text and normalize it for more complex Natural Language Processing (NLP) tasks, such as NER or Relationship Extraction. Throughout this process,

functions are applied to the raw text in order to bring it to a structure where all linguistic information needed for future computation is presented. Once this structure is available, processes like NER can then be applied. In the following sections we briefly explain some of the main functions applied to the text and their main contribution to further processing.

4.6.1.1. Tokenization

In Computer Science, tokenization is the process of demarcating sections of a string of input characters. Table 6 shows an example of such tokenization.

Input	***Output after Tokenization***
I'm afraid, but you might catch him.	'I', ''m', 'afraid', ',', 'but', 'you', 'might', 'catch', 'him', '.'

Table 6: Tokenization Example

Tokenization allows us to analyze each component separately. For example, 'I' (first token in 'I'm') can be in the subject Noun Phrase while ''m' is the head of the main verb phrase. Tokenization in many cases is the first stage of text processing. Further tasks, such as part of speech (POS) tagging and chunking (identifying the high-level constituents in a sentence), are based on the results of this process.

4.6.1.2. Sentence Boundary Detection

The sentence is considered a central processing unit in the majority of NLP tasks: POS tagging, parsing, Information Extraction, Machine Translation, Text Alignment, Document Summarization, and more. Detecting sentence boundaries mainly relies on capitalization information as well as punctuation (i.e. a sentence normally ends with characters such as a periods or question/exclamation marks). The task of sentence boundary detection is closely linked to the task of capitalized word disambiguation; the objective of which is to decide whether the first word of a sentence is a capitalized variant of a common word and hence should be treated in a case insensitive manner, or if it is a proper name and hence should be processed with respect to its capitalization. Table 7 shows an example for the sentence boundary

task with incorrect output that consists of two separate sentences and the output of the correct sentence.

Original Text	***Wrong Sentence Boundary Detection***	***Correct Sentence Boundary Detection***
The president took Mr. Smith to see his office.	- 'The president took Mr.' - 'Smith to see his office.'	'The president took Mr. Smith to see his office.'

Table 7: Sentence boundary problem example

4.6.1.3. POS tagging

Part-of-speech tagging (POS tagging), also called grammatical tagging, is the process of marking up the words in a text as corresponding to a particular part of speech, based on both its definition as well as its context, i.e. relationship with adjacent and related words in a phrase, sentence, or paragraph. The process of POS tagging assigns a POS label for each token in the input as shown in Table 8[16]. The output of this process strongly affects the performance of later tasks such as chunking, NER, or Machine Translation as errors made throughout the POS tagging process propagate to the next levels of processing.

The pipeline we use for pre-processing includes a statistical POS tagger component by Curran and Clark (2003).

[16] The pipeline we used uses the Penn Treebank tag-set. The Penn Treebank Project annotates naturally-occurring text for linguistic structure – http://www.cis.upenn.edu/~treebank/

	Output after POS Tagging		
Input	***Token***	***POS***	***Meaning***
I'm	'I'	PRP	Personal Pronoun
	''m'	VBP	Verb, non-3rd person singular present
afraid	'afraid'	JJ	Adjective
,	','	,	Comma
but	'but'	CC	Conjunction
you	'you'	PRP	Personal pronoun
might	'might'	MD	Modal
catch	'catch'	VB	Verb, base form
him	'him'	PRP	Personal pronoun
.	'.'	.	Period

Table 8: POS Tagging example

4.6.1.4. Chunking

Chunking, (also "Shallow parsing" or "light parsing") is an analysis of a sentence which identifies the constituents (e.g. noun or verb groups), but does not specify their internal structure, or their role in the main sentence. Table 9 shows an example with three types of chunks: noun phrase, verb phrase and prepositional phrase. Normally, after text processing each token in the output is labeled as either a beginning token in a chunk (e.g. B-NP refers to the first token in a noun phrase chunk), a token inside a chunk (e.g. I-NP refers to a token inside a noun phrase chunk) or O to represent a token outside of chunks. The pipeline we use for pre-processing includes a rule-based chunker by Grover and Tobin (2006).

Input	***Output after Chunking***
He reckons the current account deficit will narrow to only # 1.8 billion in September	[NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only # 1.8 billion] [PP in] [NP September]

Table 9: Chunking example

4.6.1.5. Text Processing Output

The input of the text processing stage is raw book text files. The output of the text processing tools we use is an XML file. Figure 2 shows the first three tokens in Alice in wonderland. A token is indicated by the tag <w>. Each token (depending on its type) includes information like its POS tag (the 'p' attribute), chunking information (the 'phr' attribute), tense, voice (the 'voice' attribute which may be active or passive) and more.

```
<s>
      <w phr="B-NP" headn="yes" p="NNP" c="w" l="alice" ng-
id="ng2">Alice</w>
      <w phr="B-VP" p="VBD" c="w" l="be" vg-id="vg1"
modal="no" asp="prog" voice="act" tense="past">was</w>
      <w phr="I-VP" headv="yes" p="VBG" c="w" l="begin" vg-
id="vg1" modal="no" asp="prog" voice="act"
tense="past">beginning</w> …
</s>
```

Figure 2: Text processing output

Chapter 5: Methodology

As described earlier our first task is to extract an ordered list of central characters. For example, for the book Pride and Prejudice by Jane Austin, we would like to extract an ordered list of ten central characters. In the example used in Figure 3, which is taken from Pride and Prejudice, ‘Elizabeth Bennet’ is the main character in the story, ‘Mr. Darcy’ is the second important character etc.

```
1. Miss Elizabeth Bennet
2. Mr. Darcy
3. Miss Jane Bennet
4. Mr. Bingley
5. Mr. Wickham
6. Mr. Collins
7. Mrs. Bennet
8. Lydia Bennet
9. Mr. Bennet
10.Lady Catherine De Burgh
```

Figure 3: Ordered list of central characters

For the relationship extraction task we would like to extract an ordered list of ten relationships and the relationship kinds as shown in Figure 4.

```
1. Elizabeth & Darcy - Personal
2. Elizabeth & Jane - Family
3. Elizabeth & Lydia - Family
4. Elizabeth & Mrs. Bennet - Family
5. Jane & Mr. Bingley - Personal
6. Mr. Bingley & Mr. Darcy - Personal
```

Figure 4: Ordered list of characters

The task of extracting central characters can be defined as three sub-tasks:

1. Detecting names in the text which refer to characters.
2. Resolving co-references (detecting cases were more than one name refers to the same character and unify those occurrences (e.g. ‘Lizzy and ‘Miss Elizabeth’).
3. Ordering the list of extracted characters by their importance in the story.

Our underlying hypothesis (see Hypothesis 1, section 2.2.1) is that occurrences of central characters can be found more frequently in the text.

This is true both in terms of books written in the first person and books written in the third person. In books written in the third person, we expect to find relatively more names (in relation to pronouns) while in books written in the first person we expect to find more pronouns since the pronouns and mainly personal pronouns (such as 'I', 'me', 'my' or 'myself') replace the character name. However, in both cases, whether we count the actual names or the pronouns referring to the names, we expect the main characters to appear frequently. Our experiments are all based on this frequency hypothesis. In all of the experiments, we attempt to determine the order of the lists based on different techniques to compute frequency. The experiments differ mainly in the methods they use to detect character names and resolve co-references.

In our first experiment we process our books with an existing state-of-the-art NER system (developed at the School of Informatics, University of Edinburgh (Grover and Tobin (2006); Curran and Clark (2003)). The system uses a pipeline architecture (McNeill et al. (2006)) consisting of tokenization, sentence boundary detection, POS tagging, chunking, and NER. We use the text-processing functionality of the pipeline in our other experiments (as described in section 4.6). We are interested in the performance of NER particularly on the extraction of person type entities.

Our NER system is not trained on fiction. It is trained on newswire data and aims to detect named entities of various types (e.g. location, organization, and money) while we are interested only in person entities for the scope of this project. Therefore, we consider this system as our baseline method.

5.1. Central Characters

In this section we describe four methods for identifying the main characters in a work of fiction.

5.1.1. Named Entity Recognition

The pipeline's NER component is trained to detect and label entities of a number of types, including PER (for entities representing a person such as

'Miss Bennet'), LOC (for entities representing a location such as 'London') or ORG (for entities representing an organization such as 'UN'). The NER component uses the generic tagging framework of Curran and Clark (2003) and is based on statistical methods which consider contextual linguistic information added to the text during the text pre-processing stage, particularly the part of speech and chunking information. Throughout this process, entities are detected and labeled and co-reference resolution is also applied. The output of this process is an XML file containing mark-up of all detected entities. Figure 5 shows the output in HTML, where named entities are highlighted and their IDs are displayed to the right. Additionally, we can see noun phrase chunks in square braces and verb phrase chunks in curly braces. The bottom section of the HTML file shows summaries for each entity ID where we can also see co-references[17].

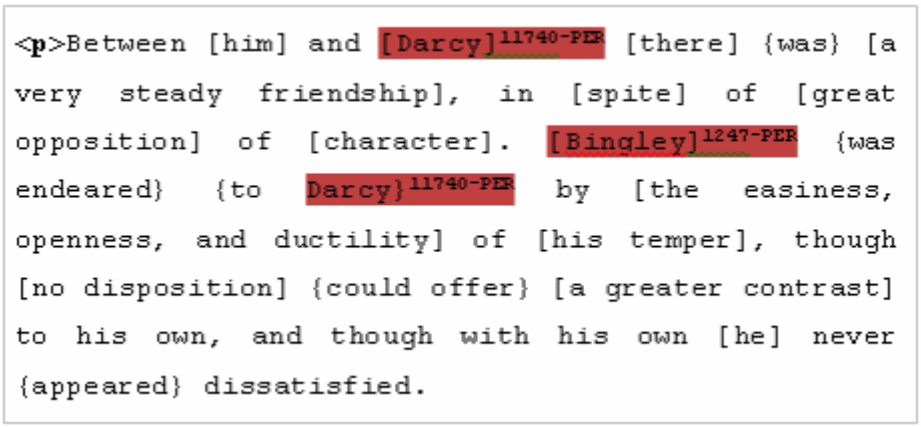


Figure 5: Named Entities in pipeline output

[17] The co-references resolution procedure here is different from the one we implement in our other two experiments.

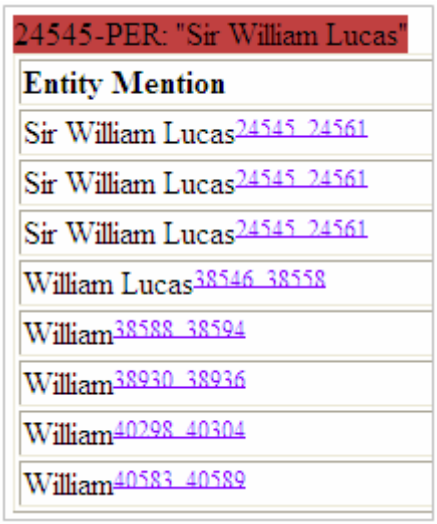


Figure 6: Named Entity lists

The example in Figure 6 shows that for the PER (person type) entity ID 24545 – 'Sir William Lucas', the process detected eight occurrences, consisting of three different names all referring to the same entity. We compute the counts for each of the entities and then generate final lists, sorted by counts, of PER, ORG and LOC entities.

To compare the output to our Gold Standard we put a threshold of ten entries on the final lists of extracted entities and use only these top ten entities to compare to our Gold Standard.

5.1.2. Co-reference Resolution

Resolving co-references is a major challenge in the task of extracting main characters. We hypothesized earlier (see section 2.2.2) that not being able to detect which names refer to the same character, will severely affect any attempt to calculate character importance and will make it almost impossible to evaluate the results or compare them to the Gold Standard. Additionally, in the genre we focus on for this research, referring to the same person using different names is very common due to the extensive use of prefixes. In Pride and Prejudice for example, the author refers to the main character, Elizabeth Bennet using eight different names: Elizabeth, Lizzy, Eliza, Miss Elizabeth, Miss Lizzy, Miss Eliza, Miss Elizabeth Bennet, and Miss Eliza Bennet

We implement a rule-based algorithm that resolves co-references. The algorithm is mainly designed to detect co-references between human entities (i.e. strings that represent names or titles of people) and it does not handle non-human entities. In our results we show how books such as Alice in Wonderland, where main characters may not necessarily be people, are affected by this.

5.1.2.1. Potential Characters

We first generate a list of objects that we consider as potential characters and feed them to our co-reference resolution algorithm. Here, we exploit the output generated during the pre-processing stage. The processed text contains a POS tag assigned to each token. We process this text in order to detect all the sequences of tokens tagged as Proper Nouns. Proper Nouns (also referred to as Proper Names) are the names of unique entities. For example, "Janet", "Jupiter" and "Germany" are Proper Nouns. Proper Nouns are capitalized in English and most other languages that use the Latin alphabet, and this is one easy way to recognize them[18]. Tokens tagged as Proper Nouns are assigned the POS tags 'NNP' for the singular form (e.g. America) or 'NNPS' for the plural form (e.g. Americas). We detect sequences of Proper Nouns and maintain them in a list along with their original order in the text. Maintaining the order is important when coming to resolve co-references since the positions in the text of two names potentially referring to the same character can help to resolve ambiguity.

5.1.2.2. The Algorithm

The co-reference resolution algorithm consists of two components: name attribute extraction and co-reference resolution. The two components are described in the following sections. For implementation details, see Chapter 6.

[18] Although the practice of capitalization is becoming less common in some domains of modern language, the texts we choose to focus our research on are written in a traditional style where names are always capitalized.

Extracting Name Attributes

Extracting attributes like first name, last name, and prefix from the terms on our list is the first step on which we can build the comparison between two names. Our goal is to generate a structure as shown in Figure 7.

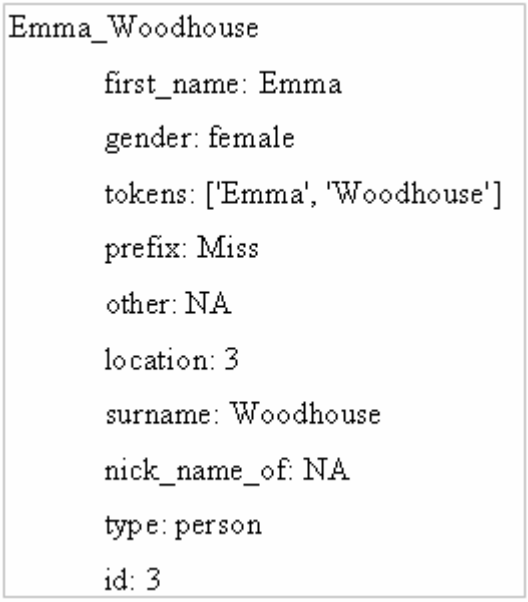


Figure 7: Name structure example for Emma

Based on this structure, we can compare two names to see if they refer to the same person. Using a set of rules, we process each term to bring the data to the above structure where we keep the extracted attributes (see section 6.3). Later on in the process, we use these properties to compare and contrast terms when trying to decide whether they refer to the same entity or not.

We use the following resources when extracting term attributes:

1. First Names list: a list of 5,167 first names of the form:
   RASHIDA 2

   RAUL 1

   RAVEN 2

   RAY 1

   Where each line contains a first name and a value: 1 for Male name, 2 for Female name, and 0 if the name is not gender specific.
2. Last Names list: a list of 18,800 last names of the form:
   GERSON

   GERST

GERSTNER

3. Prefixes list: a list of 197 prefixes of the form:
   Deacon 1

   Deaconess 2

   Patriarch 1

   Where each line contains a prefix and a value: 1 for Male name, 2 for Female prefix, and 0 if the prefix is not gender specific.

4. Nicknames list: a list of names and nicknames or abbreviations of the form:
   LIZZY ELIZABETH

   NELL NELLY

   Where the first word represents a nickname or an abbreviation and the second word represents the main name.

All lists, apart from the nicknames list, were taken from the University of Edinburgh pipeline system, which uses the same lists for the NER process. The nicknames list was manually developed by us by adding nickname information we found in our development set while reviewing results and modifying the algorithm. As future work we would like to handle nicknames automatically by measuring the distance between names or by implementing a statistical method based on nickname behavior (see section 8.1.3.1).

We use a rule based system that uses the above lists to compute attributes for each of the terms and stores them in the following dictionary-like structure:

- Tokens – A list of the tokens that the term consists of.
- Type – The type of the entity that the term represents (for this project we only classify terms as person or non-person).
- Gender – male/female.
- Prefix – The prefix preceding the name, such as ‘Mr’ in ‘Mr. Darcy’.
- First Name – The first name, such as ‘Elizabeth’ in ‘Elizabeth Bennet’.
- Last Name – The last name, such ‘Bennet’ in ‘Elizabeth Bennet’.
- Nickname of – Given a nickname like ‘Lizzy’, this field contains the name it refers to (e.g. ‘Elizabeth’).
- Other – Other names, such as ‘Angela’ in ‘Wendy Angela Darling’.

We first approach the terms by their length. Starting from terms consisting of two tokens, then three, four, and finally single tokens. At this stage we do not handle terms consisting of more than four tokens. Longer terms found in our development set do not refer to people and we make the assumption that this does not interfere with the results in a significant way. However, as future work we would like to make this process more generic and avoid handling terms by their length but instead perform the analysis by dynamically learning the term and its constituents (see section 8.1.3.1).

Handling Co-references

The input to this process is a list of terms and their extracted attributes in the order of their appearance in the book. The problem we want to resolve is the multiple occurrences of different names that refer to the same entity or character. Our algorithm consists of a rule-based system of seven rules. We apply rules according to how safe we believe they are. We look at safety as how likely the rule is to generate errors. We start with rules that detect obvious co-references, such as terms having the same first and last names, and ending with more ambiguous cases such as matching ‘Miss Bennet’ to ‘Elizabeth Bennet’ (rather than to any of her four sisters, all referred to at some point in the book as ‘Miss Bennet’). When a co-reference pair is found, we link the two terms.

Before starting to apply co-reference rules, we search our list for identical terms and link them. At this point we consider identical terms as co-references but this may not be the case (e.g. not all occurrences of ‘Miss Bennet’ are the same and refer to the same character). However, this is only a normalized starting point. As we apply the co-reference rules, we modify the links between terms if we find new co-references.

The following is an ordered list of the rules we apply throughout the co-reference resolution stage (see section 6.4.2 for a detailed list):

1. Terms with same first and last name (e.g. ‘Elizabeth Bennet’ and ‘Miss Elizabeth Bennet’).
2. Terms with same prefix and first name (e.g. ‘Miss Elizabeth’ and ‘Miss Elizabeth Bennet’).
3. Terms with same first name (e.g. ‘Elizabeth’ and ‘Miss Elizabeth’).

4. Nicknames (e.g. ‘Lizzy’ and ‘Elizabeth’).
5. Terms with same last name – males only (e.g. ‘Mr. Darcy’ and ‘Darcy’).
6. Terms with same prefix and last name (1) (e.g. ‘Miss Bennet’ and ‘Miss Jane Bennet’).
7. Terms with same prefix and last name (2) (e.g. ‘Mr. Darcy’ and ‘Mr. Fitzwilliam Darcy’).

The output of this process consists of the same list of entities but with new co-reference chains. We use this list to compute frequencies of characters in the book.

5.2. Extracting Main Characters by tf-idf

5.2.1. The tf-idf Weight

tf–idf (term frequency–inverse document frequency) is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document. The importance increases proportionally to the number of times a word appears in the document, but is offset by how common the word is in all of the documents in the collection or corpus. tf–idf is often used by search engines to find the most relevant documents to a user’s query.

The *term frequency* (tf) in the given document gives a measure of the importance of the term t_i within the particular document:

$$tf = \frac{n_i}{\sum_k n_k}$$

with n_i being the number of occurrences of the considered term i, k being another term in the document and n_k the number of occurrences of this term. The denominator is the total number of occurrences of all terms except i that appear in the document. The *inverse document frequency* (idf) is a measure of the general importance of the term (it is the logarithm of the number of all documents divided by the number of documents containing the term).

$$idf = \log \frac{|D|}{|(d_i \supset t_i)|}$$

where:

$|D|$ is the total number of documents in the corpus

$|(d_i \supset t_i)|$ is the number of documents where the term t_i appears (that is $n_i \neq 0$)

then:

$$tfidf = tf \cdot idf$$

A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; hence, the weights filter out common terms.

5.2.2. Computing tf Scores

Our input for computing the tf-idf score is the list of terms generated by the co-reference resolution component. Since tf-idf functions are designed to process single words or strings containing no spaces, we can easily use our concatenated version of the terms we are already using in our current data structure where terms like ‘Miss Elizabeth Bennet’ are represented as ‘Miss_Elizabeth_Bennet’. We compute the tf value for each entity. For example, if we count 120 occurrences of ‘entity x’ and total of 520 occurrences of all entities, the tf value for ‘entity x’ would be 0.23.

5.2.2.1. Computing idf Scores

To compute the idf score we use a collection of 1,000 books from Project Gutenberg. The books are retrieved from the LoCC category PS (‘American Literature’). For each co-reference chain we use one (random) representing term for which we compute the idf value. These random terms are not necessarily the most unique versions of each name. However, using the longest version of the name (e.g. ‘Miss Elizabeth Bennet’ instead of

'Lizzy') may be too rare. We later discuss the effect of idf values on the final tf-idf scores and how they should be taken with a grain of salt when coming to analyze the results (See section 5.2.3).

The 1,000 books are processed in the same way we process our development and test sets (they are POS tagged and searched for all sequences or Proper Nouns). We compute an idf score (see section 6.5.1) and finally, given the tf and idf values, we compute the combined tf-idf score. Our final output is a list of terms sorted by their tf-idf score as shown in Table 10.

character	***tf***	***idf***	***tf-idf***
Lizzy	0.180	4.377	0.791
Mr_Darcy	0.084	5.293	0.448
Bingley	0.049	5.986	0.294
Mr_Wickham	0.042	5.986	0.255

Table 10: tf-idf output

5.2.2.2. Handling Co-references Based on tf Scores

In addition to working with co-references detected using the method described above, we also try a different direction which is based on using the tf score as means to resolve co-references. The idea behind this method is to try a simple solution to ambiguity resolution based only on our knowledge of the importance of terms in the book, in relation to the rest of the terms in the book. This means that when a term can refer to more than one term, we use the higher tf score to determine which one of the terms it should be paired with (see Hypothesis 8A).

We use two sets of rules, 'SAFE' and 'PENDING' to determine co-references. We start with the safe rules and as we go along, we apply pending rules using information we have already computed. As a last step we resolve ambiguities using our computed tf scores. In cases where a given entity may refer to more than one entity and the conflict cannot be resolved by the rules, we prefer the reference to the entity with the higher tf score.

5.2.3. Extracting Main Characters by Frequency

When analyzing our method, which is based on the tf-idf weight to order our list of characters, we notice that this weight may not be the ideal weight to measure the importance of a term or a character in a book. This is mainly related to the idf component in the equation. By nature, the idf value is supposed to represent the uniqueness of a term in the corpus or in the collection of documents or books. The less frequent it is in the corpus, the higher the tf-idf score is. However, in terms of characters in books, the behavior seems to be quite contrary to our expectation. The more frequent a name is in other books, the more important it should be. For example, mentioning Peter Pan frequently in books other than Peter Pan, makes Peter Pan a very significant character compared to other characters in Peter Pan that are mentioned less frequently in other books. The problem this presents leads us measuring the utility of counting frequencies in the text.

This method, like tf-idf, is based on the same co-referencing algorithm described earlier. We use the same component for extracting name attributes and same co-reference resolution. However, instead of computing tf-idf scores for each entity on the list, we simply count frequencies and generate a list of terms ordered by their frequency count. An example of such a list is shown in Table 11.

Character	***Tokens***	***Count***
Lizzy	Miss Lizzy Bennet	793
Mr_Darcy	Mr Fitzwilliam Darcy	372
Jane	Miss Jane Bennet	293
Bingley	Mr NA Bingley	216
Mr_Wickham	Mr George Wickham	187

Table 11: Frequency counts output

5.3. Extracting Relationships

As explained earlier, for the task of relationship extraction we focus on single sentences written in an indirect speech style (i.e. not part of a

dialogue). We see these sentences as the clearest, and easiest to analyze in terms of detecting relationships. For this task we rely on the best results generated throughout the character extraction stage. As described in Chapter 7, our frequency count method proved to perform better than NER or tf-idf and therefore, for relationship extraction we choose the results generated by these frequency counts.

Given a list of central characters generated in the previous stage, our goal is to detect all the sentences in the book that contain exactly two main characters and determine the relationship type from descriptive words (nouns or verbs) that appear between the characters. The relationship type can be 'Family', 'Business' or 'Personal' (the same types we used for our data collection). We try a method for determining the relationship types which uses concept rules defined between keywords that may be found in the sentences and the words defining the relationship types. We later compare our results to the product of random selection of relationship types and a majority default algorithm which assigns all potential relationships in the book the most frequent relationship type according to the Gold Standard.

5.3.1. Similarity by Rules

The purpose of this method is to determine the relationship type based on a set of rules that we write. The rules define a connection or concept between keywords and relationship type words. For example, a rule can define a concept between the word 'marry' and the type 'family'. To compose the rules we mainly use keywords taken from ACE's event type definition (see section 4.2.2) but in some cases we add more rules (mainly for 'family') based on various resources found on the internet.

For each descriptive word found between characters in the sentence we identify if there is a rule relating it to one or more of the relationship types. At the end of this process we choose the relationship type we relate most words to. It is important to note that the purpose of this process is only to examine the behavior of relationships and how they appear in a sentence unit. We are aware of the fact that our set of rules are very simplistic and do not cover all possible concepts.

5.3.2. Pair frequencies

Based on our frequency hypothesis (see Hypothesis 9) we try to infer which pairs of central characters have an important relationship in the book. We do that by counting for each pair the number of sentences we find that contain only this exact pair of characters. We compute this on all the sentences in the book.

5.3.3. Comparing to Summaries

As explained earlier, we are targeting our work towards the task of book summarization and comparison. We hope that the information extracted using the methods we present here can serve the process of automatic summarization or can shed light on the complexity of the task. Therefore, as part of our experiment we apply our methods to existing summaries of our books, downloaded from study guide websites[19]. Our purpose here is to explore the appearance of characters and relationships in the summaries and how they are different in style from the full books. Earlier in this thesis we mentioned the differences between the way summaries are written compared to books (see section 2.1.1). We would like to investigate the performance of our methods on summaries since they are compact and contain most of the information we wish to extract from the books. We apply the same methods to the summaries (two baselines and a rule based system) as well as the similarity distance method below.

5.3.3.1. Similarity Distance

To measure similarity distance between the descriptive words found between character mentions in the sentence and the relationship type words, we use an existing WordNet similarity measure function. This function is included in a package of Perl modules for computing measures of semantic relatedness. In particular, it assigns a quantitative value to the relatedness of two words[20]. We apply this function to each descriptive word we find

[19] http://www.sparknotes.com and http://www.cliffsnotes.com

[20] http://search.cpan.org/dist/WordNet-Similarity/

between the two mentions of characters in the sentences, along with each relationship type word. The function returns a similarity distance score between the two given words (a number between 0 and 1). We then compute the average distance returned for each of the relationship type words and select the type with the highest score. It is important to note that this process is quite simplistic. It uses one word for each relationship type which may be general and noisy (e.g. using a word such as ‘personal’).

5.4. Summary

In this chapter, we describe the methodologies we experimented with for the tasks of character extraction and relationship extraction and which correspond to the hypotheses we listed earlier (see Chapter 2). In particular, we described three methodologies that are used to extract central characters. The first was an existing NER system to detect person entities and then order them according to their frequency. The three other methods are constructed within this project, and are based on the detection of potential strings as characters, co-reference resolution, and finally ordering the results by a frequency measure. We described two main methods: the first uses pure frequency count for the purpose of ordering, and the second is based on a tf-idf score. With the tf-idf method we experiment with two ways to resolve co-references: by contextual information and by tf value.

For the task of relationship extraction we describe the input which consists of single sentences containing exactly two characters with at least one descriptive word between the characters. We assume that these sentences define a relationship and we infer the relationship type using keyword based rules.

Chapter 6: Implementation

The purpose of this chapter is to describe in detail specific processes mentioned earlier (see Chapter 5) and to describe purely implementation aspects.

6.1. Programming Tools

All coding work is implemented using the Python programming language[21].

6.2. Text Formatting

6.2.1. Cleaning of Book Files

Prior to applying our methods to the data, we pre-process the files in order to clean them. The files downloaded from Project Gutenberg contain parts of introductory texts and closing texts. These are removed by the cleaning process. We leave chapter headings in the text as we do not expect them to interfere with the extraction process. By using regular expressions, we detect the beginning and ending of the story in the book and remove the text that exists outside the book content, whether before or after it.

6.2.2. Splitting Files for the Pipeline

Due to technical limitations, some parts of the pipeline we use as our baseline currently cannot handle large files. To work around this problem we split the full book files into smaller files sized around 100KB each[22]. We then use the smaller files for further processing. In methods other than the pipeline we use full book files.

[21] http://www.python.org/

[22] Alice in Wonderland, our smallest book, was the only file that remained as a whole and could be processed this way.

6.3. Extracting Name Attributes

6.3.1. Nicknames

Our first module adds nickname information to entities. If a term contains a nickname found on our nicknames list, we add the real name to which the nickname points to the ‘nickname_of’ field. In Figure 8, the original term detected in the text is Lizzy and the added information is ‘Elizabeth’ which the name Lizzy is the nickname of.

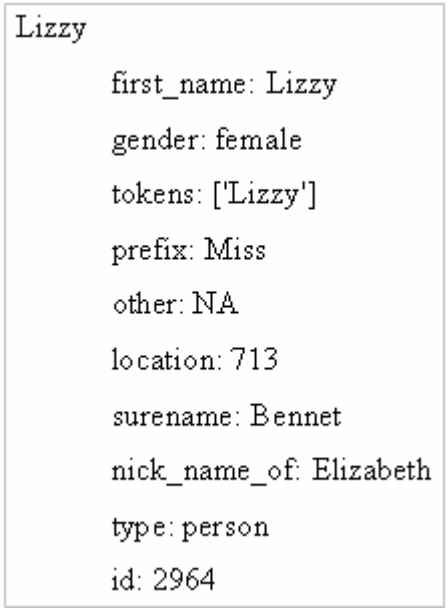


Figure 8: Example of Lizzy being a nickname for Elizabeth

6.3.2. Handling Terms by Number of Tokens

6.3.2.1. Terms Consisting of Two Tokens

We expected terms consisting of two tokens to represent names in the following possible structures:

- First Name + Last Name (such as ‘Elizabeth Bennet’)
- Prefix + Last Name (such as ‘Miss Bennet’)
- Prefix + First Name (such as ‘Miss Elizabeth’)

Hence, we first try to find the first and second tokens in all of our lists and flag them. Then we look at the flags and use the following five rules to make a decision regarding which structure most fits the term:

1. If the first token was found in the prefix list, the second token was found in the last names list but not in the first names list:

   Add found prefix as prefix, gender by the prefix gender (if available, otherwise leave as NA) and found last name as last name. Update type to ‘Person’.

2. If the first token was found in the prefix list, the second token was found in the last names list and in the first names list:

   Add found prefix as prefix, gender by the prefix gender and update type to ‘Person’.

3. If the first token was found in the prefix list and the second token was not found in any of the other lists:

   Add found prefix as prefix, gender by the prefix gender, second token as last name and update type to ‘Person’. Here we assume that an unseen name following a recognized prefix is likely to be a last name. However, as part of our future work we plan to resolve this problem in a statistical manner as well as add the new names to our lists.

4. If the first token was found in the prefix list, the second token was found in the first names list but not in the last names list:

   Add found prefix as prefix, gender by the prefix gender and found first name as first name. Update type to ‘Person’.

5. If the first token was not found in the prefix list but was found in the first names list and the second token was found in the last names list:

   Add found first name as first name, found prefix gender as prefix (assuming that the quality of the prefix gender is better than the given first name gender) and found last name as last name. Update type to ‘Person’.

6.3.2.2. Terms Consisting of Three or Four Tokens

We similarly handle terms consisting of three or four tokens. We first search for the tokens in all our external lists and then determine the most likely structure that the term represents and extract its attributes.

6.3.2.3. Terms Consisting of a Single Token

The problem with single tokens is that they do not contain enough context to help us decide unequivocally what they refer to. For example, given the term ‘Darcy’ on its own, we can not tell whether it refers to a first or last name. Therefore we start with using the information we collect up to this point. For example, if the terms ‘Mr. Darcy’, and ‘Fitzwilliam Darcy’ were previously analyzed and handled. Based on this information we can analyze ‘Darcy’ on its own and safely extract it as a last-name. We implement this by counting all the first names and last names we have already extracted. For single tokens, we first check if they have been extracted before as first or last names. If they are found to have been previously extracted as a last name, we treat them as last names and similarly for first names. If we find that they have been extracted as both, we prefer the option with the higher count. If they have not been previously extracted, we turn to our external lists. If the token is found in one of the lists we simply extract it as a first or last name. If it is found in both, we choose the option of first name. Here we assume that a single unseen token, given that it has not been pre-recognized as a last or first name, or in other words, when it first appears in the story, is more likely to represent a first name than a last name. In all successful attempts we update the term type to ‘Person’.

6.3.3. Terms Consisting of Two Tokens – Revisited

The last remaining problem is ambiguous terms consisting of two tokens. For example Mr. John where we can not tell whether John is a first or last name (in the case of Mr. Darcy we know it is a last name since if it were a first name there would be a gender conflict). Here, again, we use the same method as we use with single tokens where we check to see if and how the tokens are extracted up to this point in the process and make a decision based on this information. The decision would then be based on, for example, whether John has been extracted only, or more times, as a first name than John in Mr. John being extracted as a first name.

At this point we have all available attributes extracted from the terms and stored in the same structure we started the process from. The example in Figure 9 shows the output of this process. The dictionary-like structure is passed on to the co-reference resolution component.

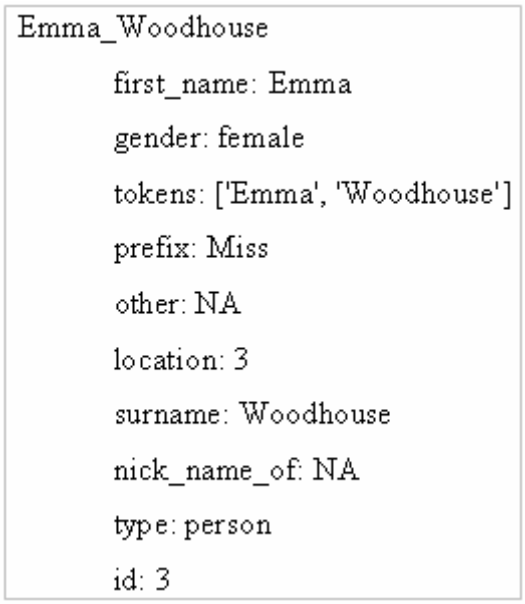


Figure 9: Final output for name component extraction

6.4. Co-reference Resolution

6.4.1. Input and Output

The algorithm takes as input a list as shown in Figure 10. It outputs a list of the same structure where the only difference is that entries referring to the same entity are assigned the same ID, thus, making a co-reference chain.

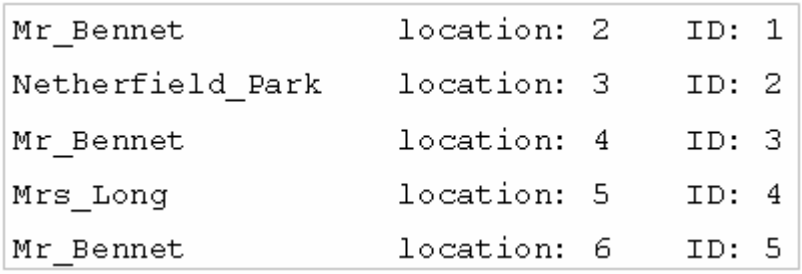


Figure 10: Co-reference input

6.4.2. Co-reference Rules

This section contains a detailed description of the co-reference rules.

Rule 1 – Terms with Same First and Last Name

The purpose of our first applied rule is to detect occurrences of terms with the same first and last name and give them the same ID regardless of the order in which they appear in the book. These are safe cases of co-reference like 'Elizabeth Bennet' and 'Miss Elizabeth Bennet'.

Rule 2 – Terms with Same Prefix and First Name

This rule handles pairs like 'Miss Elizabeth' and 'Miss Elizabeth Bennet'. Its trigger is an occurrence of a term with a first name and prefix (both different from 'NA' (Not Available)). In this case the algorithm searches the list backwards to find the closest "new"[23] term (appearing before the term being handled) which shares its first name and prefix. Both terms should also agree on the last name, meaning they cannot have different last names but any other combination is accepted. In case no term that matches these conditions is found, the algorithm switches to searching forward for terms appearing later in the story. It searches for the closest term that matches the same conditions described above.

Rule 3 – Terms with Same First Name

This rule deals with pairs like 'Elizabeth' and 'Miss Elizabeth'. It is triggered when it finds a term with a first name and no last name. The algorithm then searches backwards for the closest new term which shares the same first name. Both terms have to agree on the prefix as well (meaning that they cannot have different prefixes but any other combination is accepted). In case no term that matches these conditions is found, the algorithm switches to searching forward for terms appearing later in the story. It searches for the closest term that matches the same conditions described above.

At least one book in the set we use includes two main characters that have the same name (there are two characters named Catherine in Wuthering Heights). The rule we describe here resolves the ambiguity in the way that it only matches the current Catherine

[23] Term with a different ID

to the one appearing before her and if it cannot find one, to the one appearing after it in the text. Therefore, in terms of texts that do not necessarily use distinct first names, this algorithm should be able to resolve such ambiguity. However, this is based on the assumption that when authors introduce a second (or more) character with the same name for the first time in the story, they use unique or full names rather than just the ambiguous first or last names, in order to distinguish the new name from the original one. However, this may not be always the case. We assume that because full texts of books are relatively long and contain many occurrences of names of central characters, then even if there are ambiguous cases that can only be resolved by referent knowledge of the reader, they may be ignored.

Rule 4 – Nicknames

This rule makes use of the nickname information we add to the terms and attempts to unify cases like ‘Elizabeth’ and ‘Lizzy’ or ‘Elizabeth’ and ‘Eliza’. It is triggered when it finds a term which contains a nickname (a term which has a value in its ‘nickname_of’ field). It then searches backwards to find the closest new term whose first name is similar to the value in the ‘nickname_of’ field that the current term has. Both terms have to agree on prefix and last names (meaning that they cannot have different prefixes or last names but any other combination is accepted).

Rule 5 – Terms with Same Last Name – Males Only

This rule handles cases like ‘Mr. Darcy’ and ‘Darcy’. It is triggered when finding a term with a last name and no first name. The algorithm then searches backwards to find the nearest term that shares the same last name. Here, we introduce a new requirement. We notice that particularly in the genre we are dealing with, using last names only, mainly (if not only) refers to male characters. For example ‘Darcy’ on its own always referred to ‘Mr. Darcy’ rather than to ‘Miss Darcy’ even in cases where it appears closer in the text to Miss Darcy. Not limiting this rule to work on male characters causes errors in the co-reference resolution process. We therefore add a limitation here, requiring both terms to agree in terms of

"male-gender". If not both the terms, at least one of them has to be a male and in any event, none of them is allowed to be a female.

Rule 6 – Terms with Same Prefix and Last Name (1)

This rule handles cases like 'Miss Bennet' and 'Miss Jane Bennet'. The example we use here may seem very straight forward. However, in books dealing with a number of members of the same family (Pride and Prejudice introduces five Miss Bennets), matching the right pairs can be quite complex. Therefore it is applied as one of the last rules on our list. The rule is triggered when it finds a term with a last name and prefix and no first name. The algorithm then searches backwards in the list to find terms with no first name sharing the same last name, prefix, and gender.

Rule 7 – Terms with Same Prefix and Last Name (2)

This rule completes rule 6 but handles more ambiguous cases. The rule is triggered by terms with a last name, prefix and no first name (same as in rule 6) but in this case the algorithm is searching backwards for the closest term sharing the same last name, prefix and gender and which also has a first name. This rule aims to handle co-references of pairs like Mr. Darcy and Fitzwilliam Darcy.

Removing Stop-Words

We have noticed that due to POS tagging errors, some words are wrongly tagged as Proper Nouns, thus propagating the error onto our characters list. The common mistakes occur in all the books in our set and these terms, tagged as Proper Nouns, make it to the top of the lists since they are found frequently in the books. In order to avoid extracting these words as main characters (without changing POS taggers or fixing the current one) we instead add a stop-words list. The stop-list contains these common words (e.g. 'No', 'Yes' and 'God'). Once all the rules are applied we remove entries consisting of terms found in the stop-list.

Finalization

We apply the rules in order, for each rule we search the whole list of terms. At the end of this process we have the same list in the same structure we started with but this time with co-reference resolution

where terms referring to the same character share the same ID. Additionally, terms identified as “human” are now assigned the type ‘Person’. At this point we do not filter the list to remain with Person type of terms only since we are interested in main characters like the ‘White Rabbit’ in Alice in Wonderland or ‘Tinkerbell’ in Peter Pan which are not identified as Person. As part of our future work, we would like the task of assigning types to be fine-tuned to handle other entity types like locations, organizations or miscellaneous entities.

6.5. tf-idf

6.5.1. Computing an IDF Score

To compute the idf value we use the python module ‘advas’, which provides algorithms for advanced search. These methods are mainly used in information retrieval and linguistics. We use the provided advas function for computing an idf value for each of the terms.

6.5.2. Resolving Co-references with tf Scores

The starting point for this method is the original list of terms (consisting of sequences of Proper Nouns). Based on this list we first compute the tf value for each term on the list. We then sort the list of terms so that the most detailed terms appear at the top followed by the shorter ones that contain less information (e.g. ‘Miss Elizabeth Bennet’ appears before ‘Miss Elizabeth’ and is handled before it). The reason for doing this is to start the analysis from the “best” terms and as we move further down the list, compare “worse” terms to “better” ones. We then apply an ordered list of rules to each term. We compare terms from our sorted pool of terms with terms already existing in our “safe” list. Our analysis is done in two stages. We first handle safe cases and keep ambiguous cases in a pending list. Then, given the information collected while handling the safe cases, we handle the pending terms. If a “safe” rule can be applied to a pair of terms, we perform a unification process which includes:

- Adding the new tf score to the one already stored in the “safe” list; and
- Adding any extra attribute information found in the current term to the one in the “safe” list.

If a “pending” rule can be applied to a pair of terms, we store the pair in a pending list along with the rule type that detected them.

The following is the initial ordered set of rules that we apply to each term to indicate whether it should be matched to a term in the safe list or be stored with its match in the pending list:

1. SAFE: Terms A and B have the same first and last name (e.g. ‘Elizabeth Bennet’ and ‘Miss Elizabeth Bennet’)
2. SAFE: Terms A and B have the same first name and prefix (e.g. ‘Miss Elizabeth’ and ‘Miss Elizabeth Bennet’)
3. PEND: Terms A and B have the same first name and they are both missing a last name. Additionally, their genders and prefixes do not contradict (e.g. ‘Elizabeth’ and ‘Miss Elizabeth’)
4. PEND: Current term does not have a first name, A and B have the same last name and their genders and prefixes do not contradict (e.g. ‘Miss Bennet’ and ‘Elizabeth Bennet’)

If no rule can be applied to the current term, we append it to the safe list.

We consider the above set of rules (although it was designed using a set of texts from the 19th and 18th centuries) as generic enough to be applied to modern texts. However, they have not been tested on such texts.

At this stage we have a safe list and a pending list where we have optional pairs of terms. Our algorithm reads the terms in the pending list and handles them in two groups according to the type of rule that originally stored them in the pending list. For each term, we search the “safe” list and retrieve all possible candidates it can be matched to. At the end of this process we have two lists for two rule types, and for each of the terms in these lists we have one or more potential co-reference candidates.

At this point we can start resolving ambiguities. First, in both lists, terms that only have one candidate to which they can be matched are unified. Secondly, for terms with more than one candidate, we choose the candidate with the highest tf score and unify them. Doing this is based on

the assumption that occurrences of characters that are referred to by shorter versions of their names are likely to refer to the more important or more common character in the book. In Pride and Prejudice for example, we get ‘Darcy’ to be matched to ‘Mr. Darcy’ rather than to ‘Miss Darcy’ as ‘Mr. Darcy’’s tf score is much higher and he is mentioned more frequently in the book.

To finalize the process we repeat the steps described earlier. We remove terms consisting of stop-words and then compute the idf values and the combined tf-idf scores for each of the terms on our final list.

Chapter 7: Results and Discussion

Our algorithms produce lists of central characters and relationships ordered by importance. To evaluate them, we first set a threshold of ten to create 10-best lists, and then compare these lists to our Gold Standard. The Gold Standard consists of ordered lists of central characters and relationships that we collected during our data collection phase (see section 4.2). We compare the lists in terms of absolute order (to measure exact matching of items and their position in the list) and coverage (to measure coverage of the extracted items ignoring their exact position). Based on this comparison, we compute an average score for our test sets. Finally, we compute precision, recall and F-score values.

This chapter summarizes the most significant results in relation to the hypotheses presented earlier (see section 2.2). See Appendix D for additional detail.

7.1. Excluding Exceptions

The eight books chosen for our research are all similar in that they were all written in the 18th or 19th centuries and are considered as “periodic texts”. Nevertheless, we can categorize the eight books according to other features related to the style they are written in.

One feature that may have a great effect on the performance of our methods is the style in which the book is written; more specifically, if it is written in the first or third person. This is based on the fact that books written in the first person tend to use personal pronouns such as ‘I’ or ‘myself’ instead of Proper Names, and they often exclude the name of the central character. Our algorithms do not deal with pronouns and therefore we expect the quality of the results on books written in the first person to be generally worse. Out of our eight books, three are written in the first person: Jane Eyre, Great Expectations and Wuthering Heights.

Another characteristic of books that we think may affect the results is the type of the majority of the central characters. Most stories involve human characters. However, some stories involve other characters such as

animals or imaginary creatures. In this case we may have a problem with co-reference resolution since our co-reference resolution methods are based on name attributes like first and last names. Therefore, matching entities like ‘White Rabbit’ and ‘Rabbit’ or ‘Tink’ and ‘Tinkerbell’ may be hard.

Out of our eight books, there is only one book which we can define as “mainly involving non-human characters”. This book is Alice in Wonderland. Peter Pan involves a number of non-human characters as well but the majority of the characters are human.

Taking the different characteristics of books explained above into consideration, in this section we show our results on various samples of the books we have studied. We compute average scores on the full set of eight books, on samples excluding books written in the first person, samples excluding Alice in Wonderland (non-human characters) and samples excluding all exceptions (books written in the first person and books with mainly non-human characters).

7.2. Central Characters

When we compared the output of tf-idf with the output of our frequency count method we found that there is strong agreement on absolute order of the top two characters. The agreement on the top three characters is also quite strong (54.1%, see Appendix D, Table 45). We also find strong agreement between the two methods in terms of coverage of the complete superset of characters (over 91.2% agreement, see Appendix D, Table 46).

7.2.1. Absolute Order Results

In terms of absolute order, we find that the best results are generated when we use the tf-idf score method with co-references by contextual information. This method achieves best results on each sub-set we test our test set on, as shown in Figure 11. Here, we can also see that the results improve in accuracy as we narrow the set and they are perfect on the top character. This shows that the more prominent a character is, the easier it is to extract.

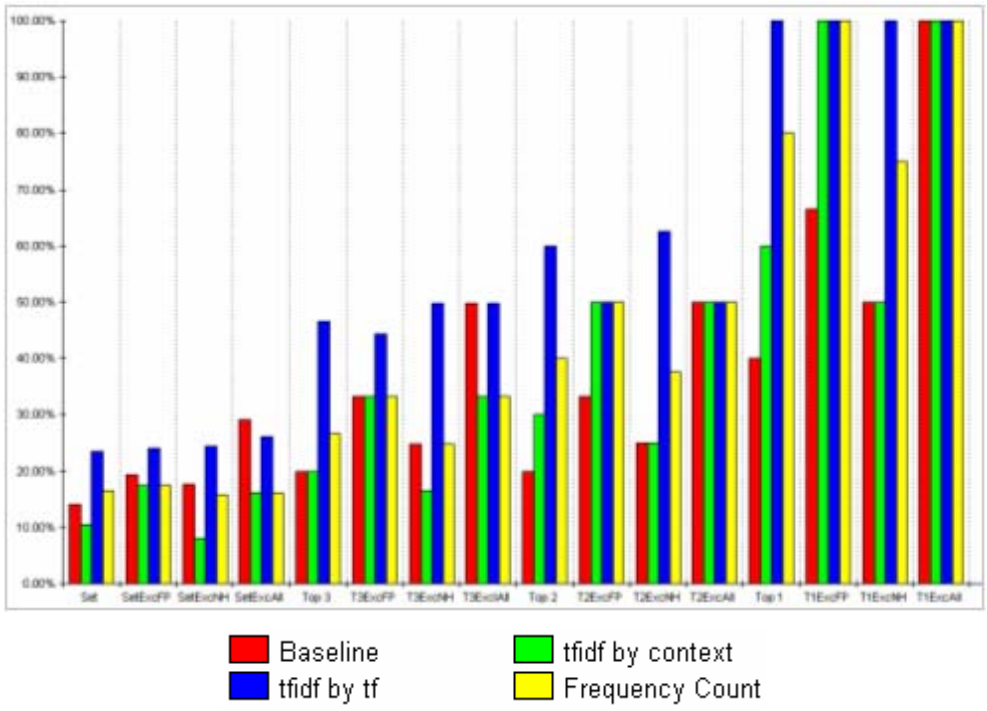


Figure 11: Absolute order results where: T1 is top 1, T2 is top 2, T3 is top 3, FP refers to First Person, and NH to Non-Human. tfidf by tf is the tf-idf method with co-reference resolution by tf scores, and tfidf by context is the tf-idf method with co-reference resolution by contextual information

7.2.2. Coverage Results

Our coverage results are particularly good, as shown in Figure 12. From all our methods, the NER which we consider to be our baseline gives the worst results. Unlike absolute order, the best results here are generated by the frequency count algorithm, which results in 100% coverage for the top three characters in the book. Moreover, we showed earlier that significant agreement is only found on the top three or less characters and that readers do not agree on the whole set of characters. In this respect, our results match the agreement level. We also discussed the current design of our data collection process and the possible effect on the evaluation (see section 4.4). We explained that the agreement level may be affected by the fact that we did not provide readers with pre-defined lists of characters. Additionally, given the relatively low number of responses, it is also possible that we do not have enough data to compute the statistics on. We therefore focus on the top three characters, but we also aim at extracting as many central characters as possible according to the current Gold Standard.

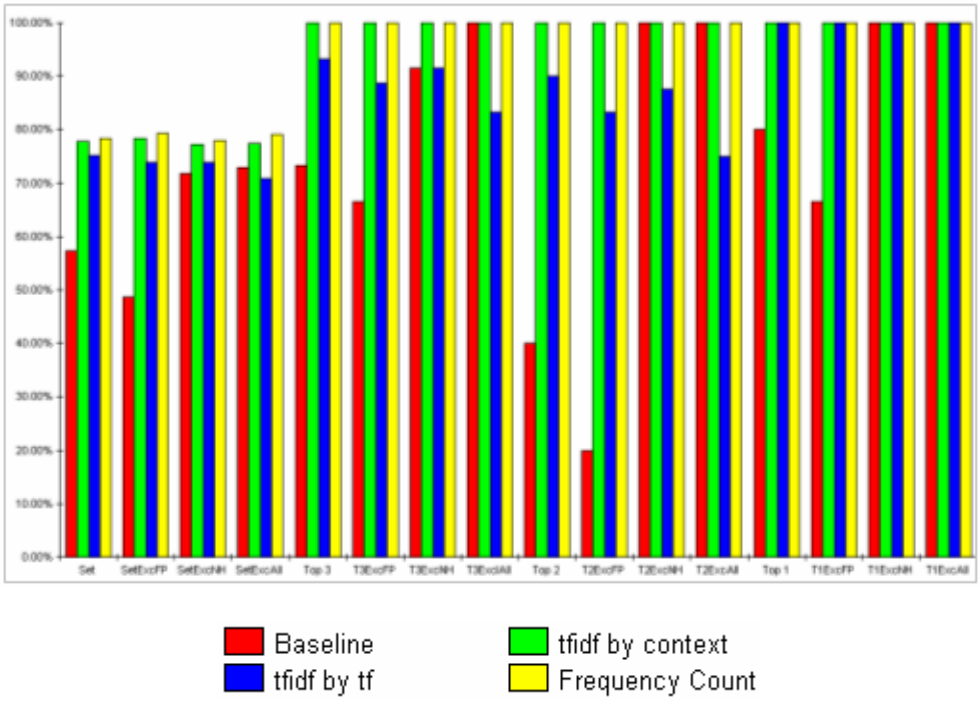


Figure 12: Coverage results where: T1 is top 1, T2 is top 2, T3 is top 3, FP refers to First Person, and NH to Non-Human. tfidf by tf is the tf-idf method with co-reference resolution by tf scores, and tfidf by context is the tf-idf method with co-reference resolution by contextual information

7.2.3. F-score

We compute precision, recall and F-score values using the three quantities given by the following formulae:

$$precision = \frac{tp}{tp + fn}$$

$$recall = \frac{tp}{tp + fp}$$

$$f - score = \frac{2 \times precision \times recall}{precision + recall}$$

where tp refers to items (characters or relationships) found and correct, fn to items found and incorrect, and fp to items not found that should have been. The F-score represents the harmonic mean of precision and recall values.

Again, our results indicate that frequency counts generate the best results as shown in Table 12.

	Precision	***Recall***	***F-score***
Baseline	50%	72.5%	58.5%
tf-idf (1)[24]	50%	77.5%	59.5%
tf-idf (2)[25]	45%	64%	52%
Frequency	55%	79%	64%

Table 12: F-score results when excluding all exceptions

Our best F-score (64%) is generated by the frequency count method when excluding all exceptions (books written in the first person and books with mainly non-human characters)[26].

7.2.4. Character Extraction Hypotheses - Revisited

We now return to the character extraction hypotheses we made in Chapter 2, to evaluate whether we have proven, disproven, or were forced to leave them untested.

Hypothesis 1: *The importance of characters within a work of fiction is directly proportional to their frequency of mention in that work of fiction.*

We showed earlier that in terms of absolute order we obtained a best score of 100% for the most important character in the book (the top character), 62.5% for the top two characters and 49.9% for the top three. In terms of coverage we showed that our best method obtained an average of 100% coverage on the top three characters. We also discussed the agreement level between readers and the nature of the data collection process. Given the combination of our coverage and absolute order results for the top three characters we can conclude that we have shown that there is a correlation between the frequency of the mentions of characters in the book and their importance. However, we cannot conclude that they are significantly proportional. In order to fully prove this hypothesis, the methods that strongly affect absolute order, such as

[24] tf-idf with co-reference resolution by contextual information.

[25] tf-idf with co-reference resolution by tf scores.

[26] The results displayed here are general and are computed for the whole set of characters.

co-reference resolution, will have to be modified and retested (See section 8.1.1).

Hypothesis 2: *Named Entity recognition (NER) techniques can be used to find a significant number of the mentions of key characters in a work of fiction.*

In terms of coverage, our NER method obtained a perfect score of 100% on the top three characters and 72.9% on the whole set of characters (when excluding all exceptions). We have proven that this hypothesis has merit, although we have not tested if the mentions found are statistically significant. Regardless of the problems we mention in section 7.2.6, the NER method did obtain promising results in finding mentions of central characters. It was not as efficient as our frequency method, however, which indicates the problem of domain transfer in NER is an important issue.

Hypothesis 3: *The most important characters in a work of fiction are the most frequently occurring person named entities.*

Here, we examined the combination of coverage and absolute order results. We mentioned earlier that in terms of coverage the NER method performed better than expected and succeeded in detecting a sufficient number of mentions of central characters in the text. In terms of absolute order, it obtained relatively good results compared to the other methods. It achieved a best score of 100% on the top character, 50% on the top two and 49.9% on the top three characters (see Appendix D, Table 42). In section 7.2.6 we discuss potential problems of our NER method, one of which is the fact that the system is trained to detect various types of entities other than person. We believe that the results can be improved using a system that is trained only on person entities in order to detect only entities of this type.

Hypothesis 4: *Identifying all the sequences of Proper Nouns in a work of fiction can provide a list of important named entities.*

For our frequency counts and tf-idf methods we used off-the-self pre-processing methods to detect sequences of Proper Nouns based on part of speech tags. Our coverage results for these two methods clearly show that we have proven this hypothesis in obtaining a perfect score of 100%

coverage on the top three characters and over 78.3% on the whole set of characters (see Appendix D, Table 43). We can safely say that detecting all the sequences of Proper Nouns provides a list of important entities and particularly central characters in a work of fiction.

Hypothesis 5: *The most important characters in a work of fiction are the most frequent proper nouns.*

Here, we look at the results in terms of coverage. We showed earlier that we obtained very high scores for coverage on the top three characters as well as on the whole set of characters when using our methods that are based on sequences of Proper Nouns, hence proving this hypothesis; the most frequent sequences of proper nouns are indeed the most important characters in a work of fiction.

Hypothesis 6: *Normalizing simple character mention frequencies by corpus-wide frequency (tf-idf) will provide a more accurate list of central characters than simple frequencies alone.*

In terms of absolute order, our tf-idf method (with co-reference resolution done by contextual information) obtained the best results with the highest score on top two and top three characters (62.5% and 49.9% respectively, see Appendix D, Table 42). However, in all our other experiments, our frequency count methods proved better. In section 7.2.6 we discuss some aspects related to the nature of the tf-idf score which may be responsible for these results.

7.2.5. Co-references

Hypothesis 7: *Calculating the frequencies of occurrences of character mentions within a work of fiction cannot be performed accurately without resolving co-referential mentions of person names.*

Our results clearly show that co-reference resolution is critical, especially in books that involve members of the same family or more than one character with the same first or last name. Table 13 shows the output of the ten most frequent names with and without co-reference resolution for the book Wuthering Heights, which involves two central characters with the same first name – Catherine. We can see that there is

no differentiation between the two Catherines and that all the occurrences of the first name alone are counted as if they refer to same character. In cases like this, it is difficult to compute a performance score since we do not know which character the name refers to. In Pride and Prejudice we found 65 occurrences of 'Miss Bennet' but we do not know which Miss Bennet in particular each of them refers to out of the possible five. Additionally, without co-reference resolution, characters like 'Mr. Heathcliff' appear more than once above our threshold of ten, thus, taking the place of other entities that should appear on the list. Finally, without co-reference resolution, the absolute order is strongly affected. In Pride and Prejudice, for example, the absolute order score drops from 100% on the top three characters and 66.6% on the whole set to 10% on the whole set and 33.3% on the top three characters and only the top character is predicted accurately.

Results without Co-reference Resolution	***Results with Co-reference Resolution***
Heathcliff	Mr. Heathcliff
Catherine	Mr. Edgar Linton
Linton	Miss Catherine Linton
Hareton	Mr. Hareton Earnshaw
Joseph	Mrs. Catherine Earnshaw
Mr Heathcliff	Joseph
Ellen	Mrs. Isabella Linton
Cathy	Ellen Dean
Edgar	Mr. Hindley Earnshaw
Hindley	Mrs. Dean

Table 13: Results before and after co-reference

Hypotheses 8A and 8B:

- **8A:** *Person names appearing in a work of fiction that may refer to more than one character are more likely to refer to a more frequent character in the book rather than to the less frequent character.*

- **8B:** *Person names appearing in a work of fiction that may refer to more than one character are more likely to refer to a character appearing closer to it in the book rather than to the less close character.*

  We showed earlier that our tf-idf method with co-reference resolution by tf score yielded the best results in terms of absolute order. However, in terms of coverage, our tf-idf method with co-reference resolution by contextual information obtained better results. We also noticed that there was a problem with resolving co-references using tf scores when dealing with books where more than one character has the same name (see example from Wuthering Heights in Hypothesis 7), where the ambiguity is not necessarily resolved by preferring the more frequent character to the less frequent one. The disambiguation decision here requires further analysis and is resolved only when considering the contextual information and in particular the location of the names in the text as stated in Hypothesis 8B. Our results show that both methods that use co-reference resolution by contextual information obtained better results. In Figure 12 we can see that these two methods obtained a perfect 100% score on the top three characters across all books. The problem of books involving more than one central character with the same name is resolved when using co-reference resolution by contextual information. Therefore, we can conclude that we have proven Hypothesis 8B and disproven Hypothesis 8A.

7.2.6. Discussion

7.2.6.1. NER

Our results show that, although it is promising, the baseline NER algorithm obtained the worst results. Using the NER system was our first choice for experimenting with detecting characters in works of fiction. We chose an existing state-of-the-art NER detection module and used it to process our data. However, in many aspects, this method was not ideal for the type of data used in this research. Firstly, the system was trained mainly on news stories. These are quite different in nature from fiction. The main difference

we noticed was the length of the input. Book texts are much longer than news stories and this presented us not only with "quality" type of problems, but also with scalability issues in some parts of the pipeline. The latter resulted in poorer performance of NER, as entities in two different processed chunks had to be unified in order to achieve full resolution. Secondly, while we were interested mainly in person entities, and possibly miscellaneous entities for non-human characters, the NER system we used was trained to detect entities of various types (person, organization, location, etc.) and it did not label entities as miscellaneous. This may have caused some characters not to be detected at all. For example, in Alice in Wonderland, the name 'Alice' was not labeled as Person, Organization or Location. Only one occurrence of ALICE (fully capitalized) was labeled as 'Organization'. Additionally, some names were assigned more than one entity type. For example, in Peter Pan, Tinker Bell was classified sixteen times as 'Person', and eleven times as 'Organization'[27]. Finally, we noticed some errors occurring in various stages of text processing such as errors in POS tagging, or NER failing to detect names containing prefixes with '.' characters such as 'Mr**.** Darcy'[27]. Therefore, it was difficult to analyze the output of the detected entities. We decided not to apply any human judgment in order to try to resolve some of the incompatibility problems described above and kept person entities (entities labeled as PER) only. Given the problems described here, and the fact that the NER system served as our baseline algorithm, the results generated by it are as expected.

7.2.6.2. tf-idf

Although our simple frequency count system obtained the best results, we expected the method based on tf-idf score to do better. We can, however, see reasons why the tf-idf score did not achieve better results. One problem that we see is with the terms we used to compute the idf value. We used a random name to represent all the names that co-refer to the same character. In particular, it was the first name, from the co-reference chain, appearing in the text. This name could be very short (e.g. 'Jane') or very long (e.g. 'Lady

[27] This issue has been corrected since we last used the pipeline.

Catherine De Burgh'). In a case such as 'Jane' which may be a very common name across books in the corpus, it can be assigned a low idf score. Only in our set of eight books we found a character called 'Jane' in two. On the other hand, longer names may be too rare and be assigned an idf score which is too high (e.g. 'Captain Peter Pan' instead of 'Peter Pan'). We assume that the best name to use is "somewhere in the middle" or possibly a combination of the frequent name components across all the terms in the co-reference chain (e.g. 'Jane Bennet' rather than just 'Jane', or 'Peter Pan' rather than 'Captain Peter Pan'). We also noticed a problem in the computation method of the idf value and in particular, we are concerned about the nature of this value. The overall tf-idf score is higher as the term is less frequent in other documents. We showed earlier that when dealing with central characters in books, the case is different (the example of Peter Pan appearing in other books). Therefore, we believe that given the current computation of idf, the tf-idf score of some characters may be lower than it should be. There is a possibility the computing a frequency score which is normalized by other books in the corpus may not be the best tool to measure importance of characters in works of fiction. Nevertheless, this may be resolved in the future by using a larger corpus for computing the idf score. We assume that with a larger corpus, the effect of the problems mentioned here may be negligible.

7.3. Relationships

As described earlier for the task of relationship extraction we used the output (an ordered list of characters) of the frequency count method which was the method obtaining the best results for the task of character extraction. We discuss here our hypotheses and the results of three methods (four on summaries). The first two are baseline methods based on random selection and majority default. The third method is based on concept rules and the forth method used on book summaries only is based on a similarity distance measure. We evaluated the results in terms of coverage.

7.3.1. Baseline Results

For the task of relationship extraction, we first applied two baseline algorithms to the texts. The first is an algorithm based on a random selection of relationship types. The second is based on a majority default selection which assigns the most dominant relationship type in the book, according to readers' responses, to all the relationships. For example if the most frequent relationship type in Pride and Prejudice according to the Gold Standard is 'Family' then by default we assign this relationship type to all potential relationships. We compute precision, recall and F-score values for each baseline algorithm.

In Figure 13 we can clearly see that the majority default baseline out-performed the random selection baseline. A best F-score of 26.7% was obtained by the random baseline, and a best F-score of 43.6% was obtained by the majority default baseline when excluding all exceptions. By looking at the performance of both our baselines, we can see that excluding exceptions (all of them or by type) clearly improved the results. Based on this, we assume that the style of the book significantly affects the ability to extract relationships.

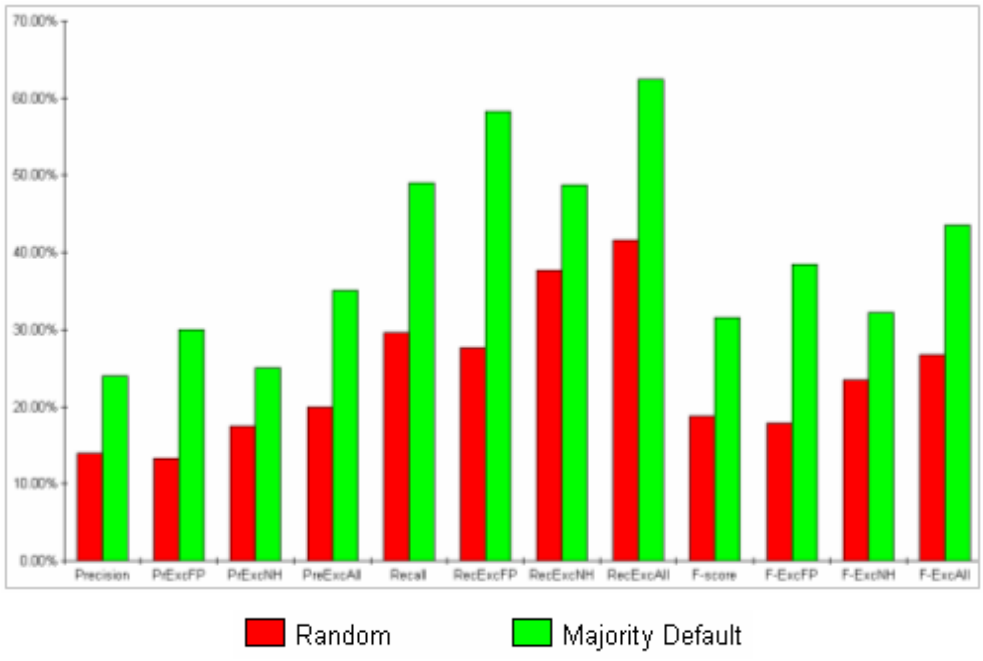


Figure 13: Baselines results, were: Pr is Precision, Rec is Recall and F- refers to F-score. FP is First Person and NH is Non Human.

7.3.2. Rule-Based Results

For our rule-based system we took as input single sentences containing exactly two characters with at least one descriptive word (a verb or a noun) between the character mentions. We applied keyword-based rules to each descriptive word with each relationship type word in order to predict the most likely relationship type. Our results show that our rule-based system performed worse than our baseline algorithms across all our sample sets. In particular, as shown in Figure 14, it obtained the lowest F-score of 20%.

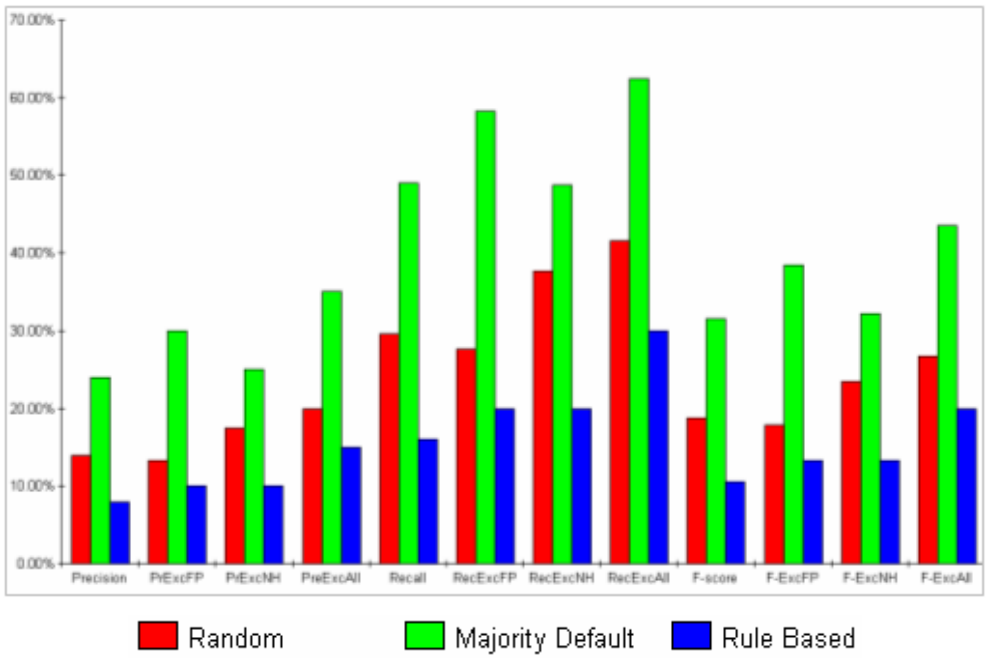


Figure 14: Baselines results, were: Pr is Precision, Rec is Recall and F- refers to F-score. FP is First Person and NH is Non Human.

It is interesting to see in Figure 14, that the majority default algorithm performed better than the random baseline and our rule-based algorithm across all the evaluated sets of books. This can serve as a base for the assumption that we can expect to find a dominant relationship type in a work of fiction.

7.3.3. Summaries

We found that the results are consistent but not better when the algorithms are applied to the book summaries. As shown in Figure 15, the majority default method obtains the best results (with an average F-score of 29.4%),

better than the random selection and our rule-based algorithms. We also found that using the WordNet similarity distance measure tended to be biased towards certain words. It returned the highest average score for the type ‘business’ for over 80% of the cases (while our lists do not contain business relationships at all). The similarity distance measure function did not label any of the pairs with the accurate relationship type.

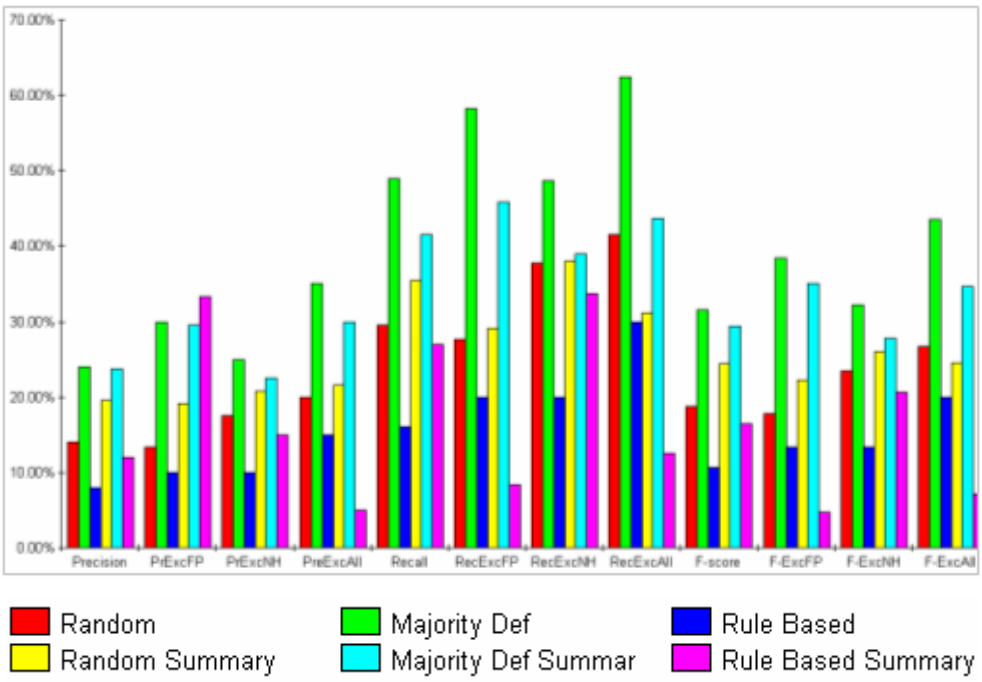


Figure 15: Relationship extraction results including book summaries

7.3.4. Relationship Extraction Hypotheses - Revisited

We now return to the relationship extraction hypotheses we made in Chapter 2, to evaluate whether we have proven, disproven, or were forced to leave them untested.

Hypothesis 9: *The importance of relationships between central characters in a work of fiction is directly proportional to their frequency of mention.*

Determining the answer here is complex. For the scope of this research we do not have the tools or the sufficient analysis capability to detect all mentions of relationships in a work of fiction. This hypothesis is related to hypothesis 11, where we can confidently say that we can determine

which pairs of central characters in the book do have an important relationship. However, proving or disproving this hypothesis requires further development and testing.

Hypothesis 10: *Important relationships and the nature of the relationship between characters in a work of fiction can be identified within the context of single sentences, and in particular sentences that are written in direct speech style and contain at least one descriptive word between character mentions.*

Our rule-based algorithm results show that we have disproven this hypothesis. As shown in Figure 14, our rule-based system obtained a best F-score of 20%. In Chapter 8, we discuss how relationships are described in works of fiction and we show that in many cases, single indirect speech sentences that contain two central characters with at least one descriptive word between then, do not necessarily describe a relationship or they may describe a relationship but not between the two characters.

Hypothesis 11: *The importance of a relationship between two characters is directly proportional to the frequency of occurrence of sentences that mention the two characters in a work of fiction, regardless of the content of those sentences.*

In our experiment, we try to infer which pairs of central characters have an important relationship in the book by counting for each pair, all the sentences this exact pair appears in. Our assumption was that the more frequent a pair of characters is in sentences in the book, the more likely there is a relationship between them. We compared the top ten pairs to the Gold Standard in terms of coverage. The results showed that our assumption has merit. For the top three relationships, we obtained a coverage score of over 74%. For the top two relationships, over 87.5% and over 75% for the top one (see Appendix D, Table 48). When excluding all exceptions we obtained a perfect coverage of 100% for all top three relationships. We did not however, measure the results in terms of absolute order, and therefore we cannot explicitly state that the importance of the relationship is directly proportional to the frequency of sentences containing the pair of characters.

Hypothesis 12: *Analyzing the descriptive words appearing between two character mentions in individual indirect speech sentences will provide sufficient information for indicating what type of relationship exists between those characters.*

We have disproven this hypothesis using our rule-based algorithm which obtained a best F-score of 20% as shown in Figure 14. We also found that the similarity distance measure is generally biased towards the 'family' relationship type. As we discuss later in this chapter, based on our in-depth analysis of the first four chapters in Pride and Prejudice, we believe that relationships cannot be identified with good accuracy based on single sentences only (see section 7.3.5).

Hypothesis 13: *By computing an average of the similarity distance between the content words found in the context of two character mentions and the content words used to describe our main relationship types, we can determine the most likely relationship type between those two characters.*

As described in section 7.3.3, the similarity distance measure algorithm did not assign any relationships with the accurate relationship type. Therefore, given our simplistic methodology of measuring similarity by general keywords, we have disproven this hypothesis.

Hypothesis 14: *We can define a set of keywords which will be an indication of particular relationships, and use these to accurately predict relationship types if these keywords appear in the context of two characters in a work of fiction.*

Earlier we discussed the problems we found in trying to infer relationships and their types from single sentences, and they are also discussed in detail in section 7.3.5. Our rule-based algorithm results indicate that using only the descriptive words between the character mentions is not sufficient with best F-score of 20% (see Figure 14). Therefore, we have disproven this hypothesis. We later discuss the implications of this and propose new directions for future work in regards to detecting relationship mentions and their types in works of fiction (see Chapter 8).

7.3.5. Discussion

7.3.5.1. Sentences Defining Relationships

Our relationship extraction results demonstrated low accuracy, mainly for the task of extracting the relationship type. As we discuss later, in Chapter 8, based on our analysis of the first four chapters in Pride and Prejudice we can say that relationships cannot be significantly identified based on single sentences. Moreover, single sentences that contain exactly two characters do not necessarily define relationships in general or relationships between the two characters in particular. In this respect, for some sentences, our algorithms were forced to detect relationships in sentences that do not describe them. Furthermore, our coverage results of methods that try to infer the nature of the relationships do not show that the relationship type can be extracted from such sentences.

7.3.5.2. Books Written in the First Person

One of the problems we noticed was in books written in the first person. For the task of character extraction we noticed that when excluding all exceptions the results improve. However, with relationship extraction, the performance on books written in the first person is significantly worse. In our worst case, for the book 'Jane Eyre', no pairs containing the character 'Jane' appeared in the top ten counts of pairs[28].

7.3.5.3. Keyword-based Rules

When we examined the rules we used for our rule-based algorithm we found that rules relating keywords to the word 'family' were defined more thoroughly and were more detailed than the rest of the rules. We believe that this is due to the nature of the domain of 'family', being rather concrete. We could find many words that could quite safely be related to the type 'family' (e.g. 'sister', 'marriage', or 'wife'). For domains such as 'business' or 'personal', rules had to be more abstract and therefore, more complex to define. In general, we believe that the method of using rules based on single

[28] Jane is the person who tells the story

words to infer relationship types is very simplistic and should be replaced with more sophisticated methods and we propose a number of possible directions in Chapter 8.

Chapter 8: Conclusions and Future Work

8.1. Conclusions

8.1.1. Character Extraction

In this thesis we have shown that it is possible to extract central characters relatively well, especially in terms of coverage. We found that all of our methods performed better than our NER baseline. However, in terms of absolute order we found that the NER method performed better on some samples of the set. We believe that this was related to our co-reference method and in particular to the algorithm which was responsible for extracting name components. The algorithm that performed the same task in the NER system was more mature and handled names in a more generic manner. Our hypotheses for character extraction are shown again in Table 14.

Hypothesis 1	The importance of characters within a work of fiction is directly proportional to their frequency of mention in that work of fiction.
Hypothesis 2	Named Entity recognition (NER) techniques can be used to find a significant number of the mentions of key characters in a work of fiction.
Hypothesis 3	The most important characters in a work of fiction are the most frequently occurring person named entities.
Hypothesis 4	Identifying all the sequences of Proper Nouns in a work of fiction can provide a list of important named entities.
Hypothesis 5	The most important characters in a work of fiction are the most frequent proper nouns.
Hypothesis 6	Normalizing simple character mention frequencies by corpus-wide frequency (tf-idf) will provide a more accurate list of central characters than simple frequencies alone.
Hypothesis 7	Calculating the frequencies of occurrences of character mentions within a work of fiction cannot be performed accurately without resolving co-referential mentions of person names.
Hypothesis 8	Person names appearing in a work of fiction that may refer to more than one character are more likely to refer to a more frequent character in the book rather than to the less frequent character. (8A); or
	Person names appearing in a work of fiction that may refer to more than one character are more likely to refer to a character appearing closer to it in the book rather than to the less close character. (8B)

Table 14: Character Extraction Hypotheses

We found that the importance of characters is directly proportional to their frequency in the text, thus proving hypothesis 1. We deduce this from the high results we obtained for coverage (100% on the top three characters) and the relatively good results we obtained for absolute order (in 100% of the cases we predict the most important character in the book and we obtain 49.9% success on the top three characters). This shows that inferring the order of importance of characters in the text from frequencies is an acceptable method, as stated in hypotheses 4 and 5.

We have not proven hypotheses 2 or 3, however, we cannot disprove them on the basis of our results. We have also shown that hypothesis 6 may not be true, although again, we cannot say we have disproven it.

We found that resolving co-references is critical for character extraction, particularly in terms of absolute order, hence, proving hypothesis 7. Co-reference resolution is more critical in books that contain family members or more than one character with the same name. Our co-reference resolution method based on using contextual information, proved better than the co-reference resolution based on term frequency (the tf value). This is especially noticeable in books with more than one character with the same name. Hence we proved hypothesis 8B and disproved hypothesis 8A.

8.1.2. Relationship Extraction

Our hypotheses for relationship extraction are shown again in Table 15.

Hypothesis 9	The importance of relationships between central characters in a work of fiction is directly proportional to their frequency of mention.
Hypothesis 10	Important relationships and the nature of the relationship between characters in a work of fiction can be identified within the context of single sentences, and in particular sentences that are written in direct speech style and contain at least one descriptive word between character mentions.
Hypothesis 11	The importance of a relationship between two characters is directly proportional to the frequency of occurrence of sentences that mention the two characters in a work of fiction, regardless of the content of those sentences.
Hypothesis 12	Analyzing the descriptive words appearing between two character mentions in individual indirect speech sentences will provide sufficient information for indicating what type of relationship exists between those characters.
Hypothesis 13	By computing an average of the similarity distance between the content words found in the context of two character mentions and the content words used to describe our main relationship types, we can determine the most likely relationship type between those two characters.
Hypothesis 14	We can define a set of keywords which will be an indication of particular relationships, and use these to accurately predict relationship types if these keywords appear in the context of two characters in a work of fiction.

Table 15: Relationship Extraction Hypotheses

Our rule-based method did not out-perform our baseline algorithms. From the three algorithms we applied to the text to try to infer the relationship type, the majority default algorithm obtained the best results. This may show that a book can be characterized by its dominant relationship type but it needs further research.

We found that the existence of important relationships between two characters can be predicted by the frequency of occurrence of sentences containing the two characters, giving merit to hypothesis 11. Hence, we can infer that two main characters found in the text are likely to have an important relationship (without inferring the relationship type) based on the

pair's frequency in all types of sentences found in the text. However, we cannot say that the importance of the relationship is directly proportional to the frequency. Hence, we cannot prove hypothesis 11, however, we cannot disprove it at this point either.

The results obtained by our rule-based algorithm were quite low, and in particular they were lower than both of our baseline algorithms. Hence, we can infer that analyzing the descriptive words between two mentions of characters in single sentences may be too simplistic, hence, disproving hypotheses 12. Moreover, by showing that trying to predict the nature of the relationship from the descriptive words using keyword-based rules or similarity distance measure techniques does not achieve accurate results, we disproved hypothesis 13 and 14.

Earlier we discussed the relatively low results of our rule-based algorithm and the complexity of detecting relationship mentions in works of fiction. We showed how relationships can be described using different linguistic methods and writing styles. It was difficult to judge the importance level of detected relationships mainly because we could not significantly prove hypotheses 10, 13 and 14, and particularly hypothesis 10 which discusses the efficiency of single sentences. Therefore, we disproved hypothesis 9.

8.1.2.1. Relationship In-depth Analysis

The way relationship mentions are described in books is not consistent and the task of detecting those relationships is extremely complex. When manually analyzing the first four chapters of Pride and Prejudice we found that:

a. There are no obvious patterns.
b. Relationships are not necessarily defined by indirect speech sentences containing exactly two characters and descriptive words. For example, the sentence in Figure 16, taken from Pride and Prejudice, contains two main characters and descriptive words between them. However, this sentence does not define a relationship between these two characters. Hence, hypothesis 10 is most likely to be false.

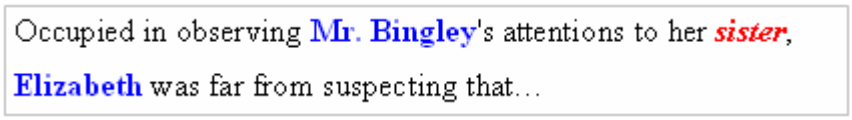


Figure 16: Sentence example where the inferred relationship is incorrect

c. Relationships can be described using single sentences, or in a larger scope such as paragraphs. Hence, hypothesis 10 can not be proved since sentences alone are not sufficient.

d. Relationships can be described in sentences containing mentions of two central characters but also in sentences containing more than two central characters.

e. Relationships are often mentioned in direct speech sentences or in combinations of direct speech and indirect speech sentences.

f. Relationships are often described using pronouns rather than the names of the characters.

8.1.3. Future Work

8.1.3.1. Character Extraction

For the task of character extraction we would like to suggest the following:

1. *Improve the algorithm that is responsible for extracting components of names and in particular make it more generic.*

   This algorithm should not depend on the length of the name and should be based on a method which parses the name and detects its features dynamically rather than by applying a finite set of rules to it. The pipeline we were using in this project contains such component and we plan to incorporate it in our system.

2. *Handle cases of nicknames and abbreviations in a more efficient manner.*

   We tried resolving nicknames using the Levenshtein Distance[29] measure but this resulted in noisy output as the algorithm gives better

[29] The **Levenshtein distance** or **edit distance** between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character.

scores to short distances although these are not always the case when dealing with nicknames (e.g. Levenshtein Distance prefers matching ‘Tink’ to ‘Think’ than ‘Tink’ to ‘Tinkerbell’). For this task, we plan on experimenting with an existing system for identifying abbreviation definitions (Schwarts and Hearst (2003)).

3. *Add the ability to dynamically extend our lists with unseen terms we detect in the text.*

   For example if we find a token that is not in the last name list but we make a decision that it is a last name, this token should be added to the lists.

4. *Refine the tf-idf score.*

   We believe that this score can obtain better results than pure frequencies. However, it needs to be fine-tuned mainly in terms of the idf value. This can be done by improving the way terms are searched for in the corpus as well as adding more books to the corpus to see if the idf score does not increase significantly when found in other books. We believe that improving the tf-idf score can have great effect on the extracted characters list, especially in terms of absolute order.

5. *Improve the way we handle types of names that we extract.*

   For this project we focused on person names only. However, for future tasks we would like to be able to determine for each name we detect in the text, more specifically, what its type is (e.g. if it is a name of a person, a location, organization, or a miscellaneous type).

We believe that these modifications can significantly improve the process of character extraction and in particular the results of the co-reference resolution.

8.1.3.2. Relation Extraction

For the task of relation extraction we would like to perform the following:

1. *Incorporate advanced patterns for relationship extraction.*

   In particular we would like to investigate relationship extraction using direct speech sentences as well as combinations of these sentences and indirect speech sentences. We would like to detect the characters involved in speech acts, for example who the speaker is, who the

receiver is and what other characters are witnessing the speech act. We would also like to add the ability to detect phrases such as “Mr. and Mrs. Bennet” and to infer from them that there is a relationship between Mr. Bennet and Mrs. Bennet and that the relationship is very likely to be of the ‘family’ type.

2. I*nfer relationships by transitivity.*

   Based on relationships that are already extracted we would like to add the ability to infer new relationships based on transitivity. For example, if we already know that Mrs. Bennet is the mother of Elizabeth and Elizabeth is Jane’s sister, we can predict that Mrs. Bennet is Jane’s mother.

3. *Use contextual information outside sentence boundaries.*

   For the scope of this research we only use the words that appear between mentions of two characters in single sentences. We would like to expand the context we consider and use the words outside the mentions of characters as well as the context outside the sentence boundaries.

4. *Consider sentences containing more than two characters.*

   In this research we only use sentences containing exactly two central characters. However, sentences with more than two characters can describe relationships too and should be analyzed.

5. *Add pronoun resolution.*

   We believe that in order to obtain good performance of relationship extraction from fiction, using names of characters only is not enough. In many cases relationships are described in sentences containing names and pronouns or only pronouns. This is especially relevant to books written in the first person where the first character is referred to using first person pronouns. We expect pronoun resolution to significantly improve the results of relationship extraction as well as character extraction.

6. *Create an annotated corpus.*

   We believe that in order to provide a good solution to the problem of relationship extraction from fiction, statistical and machine learning methods should be incorporated. In order to be able to use these

methods, an annotated corpus should be created. This corpus should contain texts of books with annotation of central characters, relationships between them and also main story events. Based on this corpus, machine learning methods can then be trained and applied to new texts.

8.1.3.3. Event Extraction and Entity Attributes

Returning to the motivation behind this research, we see this work as a step towards creating a solution for automatic summarization and book comparison. In order to be able to generate good summaries or to compare books based on various features, more information should be extracted. Firstly, we would like to extract further information about characters and relationships. For example, the ages of central characters, their gender, profession, or how their relationships develop over time. Secondly, we would like to extract main story events in their order of occurrence over time. Based on existing summaries we investigated, we believe that this kind of extracted information will allow us to generate good and sufficient summaries. Additionally, extracting events such as weddings, births, murder acts, or violent conflicts can tell us a lot about the nature of a book. This information can then be used for categorization and aggregation. It will also enable the enhancement of recommendation methods and in particular adding recommendations based on novel features such as recommending a book whose story is set in Edinburgh in the 1980's to a reader who is currently viewing Trainspotting's product detail page.

References

[1] E. Agichtein, L. Gravano. Snowball: Extracting Relations from Large Plain-Text Collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, 85-94, 2000.

[2] C. Barbu and R. Mitkov. Evaluation tool for rule-based anaphora resolution methods. In *Proceedings of ACL'01*, 34-41, Toulouse, France, 2001.

[3] M. Becker, W. Drozdzynski, H.U. Krieger, J. Piskorski, U. Schäfer, F. Xu. SProUT - Shallow Processing with Typed Feature Structures and Unification. In *Proceedings of the International Conference on NLP (ICON 2002)*, Mumbai, India, December 18-21, 2002.

[4] T. Betts. Using Text Mining to Place Books into an Ontology. MSc thesis. University of Edinburgh, 2006.

[5] K. Bontcheva, M. Dimitrov, D. Maynard, V. Tablan, H. Cunningham. Shallow Methods for Named Entity Coreference Resolution. In *Proceedings of TALN 2002*, Nancy, France, 2002.

[6] M. Collins. Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of the 35th annual meeting of the ACL*, 16-23, Madrid, Spain, 1997.

[7] A. Culotta and J. Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of ACL-2004*, Barcelona, Spain, 2004.

[8] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, 2002.

[9] J. R. Curran and S. Clark. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the 11th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-03)*, 91-98, 2003a.

[10] J.R. Curran and S. Clark. Language Independent NER Using a Maximum Entropy Tagger. In *Proceedings of the 7th Conference on*

Natural Language Learning (CoNLL-03), 164-167, Edmonton, Canada, 2003b.

[11] G.F. DeJong. Fast Skimming of News Stories: The FRUMP System. PhD thesis. Yale University, New Haven, CT, 1978.

[12] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named Entity Recognition through Classifier Combination. In *Proceedings of CoNLL-2003*, 168-171, Edmonton, Canada, 2003.

[13] J. Goldstein , V. Mittal , J. Carbonell , M. Kantrowitz. Multi-document summarization by sentence extraction. In *Proceedings of ANLP/NAACL Workshop on Automatic Summarization*, 40-48, Seattle, Washington, 2000.

[14] C. Grover and T. Richard. Rule-Based Chunking and Reusability. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy, 2006.

[15] B. Hachey and C. Grover. Extractive Summarisation of Legal Texts. *Artificial Intelligence and Law: Special Issue on E-government*, to appear.

[16] K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. Description of the LaSIE system as used for MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.

[17] S. Lappin and H. Leass. An Algorithm for Pronominal Anaphora Resolution. *Computational Linguistics*, 20(4):535–561, 1994.

[18] M. Marcus, B. Santorini and M. Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313-330, 1993.

[19] R. McDonald. Extracting Relations from Unstructured Text. Technical Report: MS-CIS-05-06, draft, unpublished.

[20] F. McNeill, H. Halpin, E. Klein and A. Bundy. Merging Stories with Shallow Semantics. *Knowledge Representation and Reasoning for Language Processing Workshop at the European Association for Computational Linguistics (EACL) conference (KRAQ 2006)*, Genoa, Italy, 2006.

[21] A. Mikheev, C. Grover and M. Moens. Description of the LTG System Used for MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.

[22] S. Miller, H. Fox, L.A. Ramshaw, R.M. Weischedel. A Novel Use of Statistical Parsing to Extract Information from Text. In *Proceedings of 1st Meeting of the North American Chapter of the ACL*, 226-233, 2000.

[23] R. Mitkov. Robust Anaphora Resolution with Limited Knowledge. In *Proceedings of COLING' 98/ACL'98*, 1998.

[24] K. Müürisep and M. Pilleriin. ESTSUM - Estonian newspaper texts summarizer. In *Proceedings of The Second Baltic Conference on Human Language Technologies*, 311-316, Tallinn, April 4-5, 2005.

[25] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4), 380-393, 1997.

[26] L. Rau and P. Jacobs. Creating segmented databases from free text for text retrieval. In *Proceedings of the 14th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 337-346, New York, 1991.

[27] A. Schwartz and M. Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Proceedings of the Pacific Symposium on Biocomputing*, 2003.

[28] N. Sutterer. Named Entity Recognition. *Seminar on Text Mining and Ontology Learning*, lecture notes, Freiburg University, June 2006.

[29] D. Zelenko, C. Aone, A. Richardella. Kernel Methods for Relation Extraction. *Journal of Machine Learning Research*, 1083-1106, 2003.

[30] G.D. Zhou, J. Su. Named Entity Recognition using an HMM-based Chunk Tagger. In *Proceedings of the 40th Annual Meeting of the ACL*, 473-480, Philadelphia, PA, 2002.

Appendix A: Data Collection Website

Site Structure

The site consists of three pages that are described in the following sections.

Welcome Note and Titles List

Here, we provided information about the project and some guidelines for filling the form. The list of titles appears below the welcome note. The list consists of links where the user can click on a link which leads him to the book form. Users can also access the full book texts by clicking on the relevant links.

Figure 17: Welcome note and title list

Form

The form is the main page where users are requested to fill in information about the book. The form is divided into five sections:

General

This section consists of the title of the book, a link to the book text and a part where users are requested to state whether they have read the book or seen the film (or both). This is done by ticking check-boxes and the default selection is ‘Read the book’:

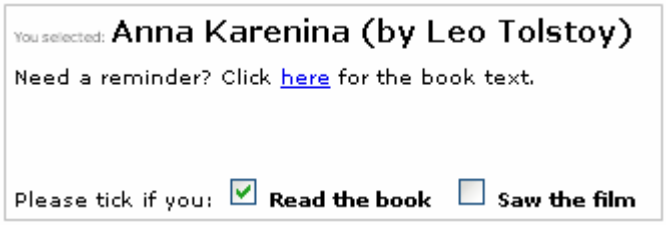


Figure 18: Form Page – General

Central Characters

In this section users are requested to type in names of central characters (three mandatory names and four optional ones) where the order represents the importance of the character in the book:

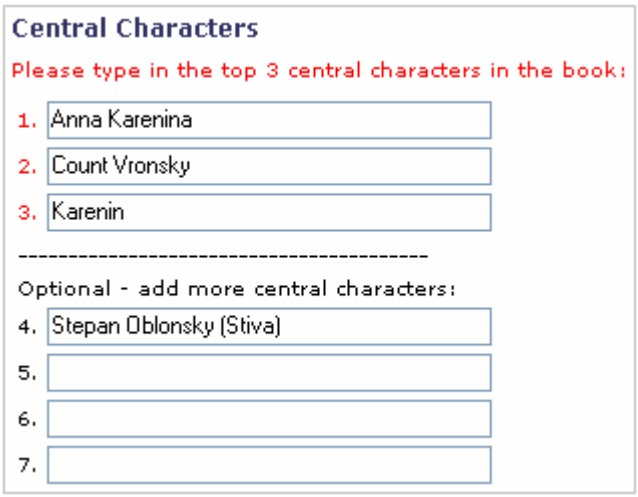


Figure 19: Form Page – Central Characters

Relationships

In the Relationships section, users are requested to specify pairs of central characters using drop-down lists (containing numbers from 1 to 7). The numbers correspond to the central characters they define in the Central Characters section. Then users have to choose the relationship type from a drop-down list and specify the relationship in the available text-box:

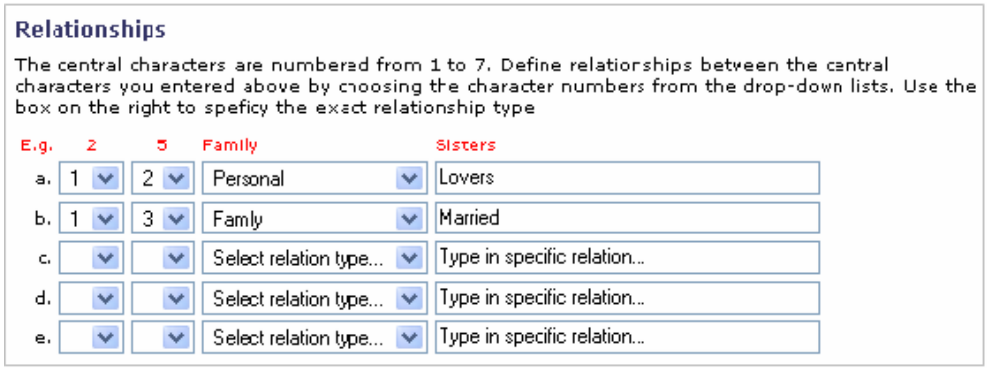


Figure 20: Form Page – Relationships

Events

In this section users have to choose an event type from a drop down list and then choose one or two central characters from the drop-down lists that correspond to the central characters they define previously. Then they have to specify the event in the available text-box:

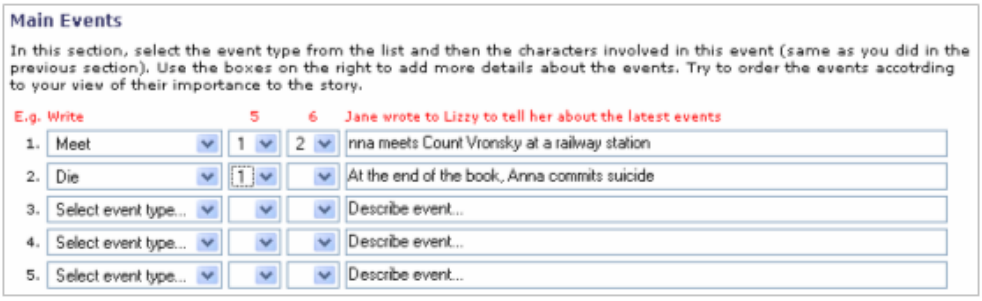


Figure 21: Form Page – Events

Additional Question

The purpose of this question is to try and collect readers' ideas on how books can be recommended. This section has no direct connection to the immediate goals of the project but we were hoping to get some insight using

readers' views on how extracted information can be used improve recommendation systems.

Figure 22: Form Page – Extra Information

At any stage users can click on the Back button at the top of the page and return to the main page to choose another title. Clicking on the Send button at the bottom of the page sends the filled in information to us by email.

Feedback

This page is displayed once the form is sent. Here users can click the 'Choose another title' button to return to the main page and choose another title from the list.

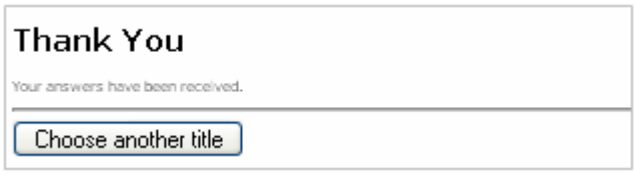


Figure 23: Feedback page

Potential Participants

We request readers' participation by sending emails to various lists within the University of Edinburgh [30] as well as to two external lists:

- Corpora List[31] – A list open for information and questions about text corpora such as availability, aspects of compiling and using corpora, software, tagging, parsing, bibliography, conferences etc.

[30] Mainly the School of Informatics and the School of Philosophy, Psychology and Language Sciences.

- ELSNET List[32] - a Europe-based forum dedicated to human language technologies. ELSNET operates in an international context, and deals with all human communication research areas related to language and speech across discipline boundaries.

The Email

Subject: Information Extraction from Fiction: Collecting Test Data

Dear all,

My name is Sharon Givon and I'm an MSc student in the Speech & Language Processing program at the University of Edinburgh. My dissertation project deals with extracting information from fiction (with Amazon.com): central characters, relationships between them and main story events. Unfortunately, no annotated corpus is available for that purpose, and this is where I need your help.

If you are willing to help, you will find in the attached link a list of very famous books. If you think you are familiar enough with a story (either from reading the book or watching the film), please click on its link to fill in some information about it. You will be asked to fill in names of central characters, relationships between them and short description of main events. If you need to refresh your memory you can use the links to the actual book texts.

Collecting this information is crucial to my project and will hopefully be useful for more researchers. I would extremely appreciate it if you dedicated a few minutes to it.
Do not feel like you have to fill in information for the whole list of titles: a few books would be great but even one book would be well appreciated.

---

31 http://torvald.aksis.uib.no/corpora/

32 http://www.elsnet.org/index.html

Here's the link:
http://sgivon.tripod.com/Index.html

Feel free to email me with questions or comment.

Regards,
Sharon.

Validation

It is important to state that due to time constraints we do not have a validation mechanism in any of the sections of the form. This means that users can leave whole sections unfilled, define relationships on a single character etc. We rely on the fact the users read the guidelines provided both in the body of the email requesting their participation and on the main page of the data collection website. In practice, almost all the users fill in the forms according to the guidelines. When we finalize the collected lists to be used as Gold Standard we excluded invalid responses (see Appendix C).

Appendix B: Ordering Methods

Central Characters

Book	***Method 1 (Median value)***	***Method 2 (Average score and number of users)***
Pride and Prejudice	1. Miss Elizabeth Bennet (1) 2. Mr. Darcy (2) 3. Miss Jane Bennet (3) 4. Mr. Bingley (4) 5. Mr. Wickham (6) 6. Lydia Bennet (7) 7. Mr. Collins (100) 8. Mrs. Bennet (100) 9. Mr. Bennet (100)	1. Miss Elizabeth Bennet 2. Mr. Darcy 3. Miss Jane Bennet 4. Mr. Bingley 5. Mr. Wickham 6. Mr. Collins 7. Mrs. Bennet 8. Lydia Bennet 9. Mr. Bennet
Alice in Wonderland	1. Alice (1) 2. Rabbit (2) 3. Queen (3) 4. Mad hatter (4) 5. Cheshire Cat (5)	1. Alice 2. Rabbit 3. Queen 4. Mad hatter 5. Cheshire Cat
Emma	1. Emma Woodhouse (1) 2. Mr. (George) Knightley (2) 3. Harriet Smith (3.5) 4. Jane Fairfax (4.5) 5. Frank Churchill (5) 6. Rev. Elton (53) 7. Mr. Woodhouse (53.5) 8. Miss Bates (100)	1. Emma Woodhouse 2. Mr. (George) Knightley 3. Harriet Smith 4. Frank Churchill 5. Jane Fairfax 6. Mr. Woodhouse 7. Rev. Elton 8. Miss Bates
Great Expectations	1. Pip (1) 2. Estella (2) 3. Miss Havisham (3) 4. Magwich / Abel	1. Pip 2. Estella 3. Miss Havisham 4. Joe (Gargery)

	Magwitch (4) 5. Joe (Gargery) (4.5) 6. Mr. Jaggers (53.5) 7. Biddy (53.5) 8. Herbert (53.5)	5. Magwich 6. Herbert 7. Mr. Jaggers / Biddy
Jane Eyre	1. Jane Eyre (1) 2. Mr. Rochester (2) 3. Bertha (Rochester's wife) (5) 4. Helen Burns (100) 5. St. John Rivers (100) 6. Blanche Ingram (100) 7. Mrs. Reed (100)	1. Jane Eyre 2. Mr. Rochester 3. Bertha (Rochester's wife) 4. St. John Rivers 5. Blanche Ingram / Mrs. Reed 6. Helen Burns
Little Women	1. Jo March (1) 2. Beth March (2) 3. Amy March (3) 4. Meg March (4) 5. Laurie (5) 6. Mr. March (6)	1. Jo March 2. Beth March 3. Meg March / Amy March 4. Laurie 5. Mr. March
Peter Pan	1. Peter Pan (1) 2. Wendy (Darling) (2) 3. Captain Hook (3) 4. Tinkerbell (4) 5. John (Darling) (52.5) 6. Michael (Darling) (6)	1. Peter Pan 2. Wendy 3. Captain Hook 4. Tinkerbell 5. Michael 6. John
Wuthering Heights	1. Heathcliff (1) 2. Catherine Earnshaw (2) 3. Hindley Earnshaw (4) 4. Edgar Linton (52.0) 5. Nelly Dean (52.2) 6. Lockwood (53)	1. Heathcliff 2. Catherine Earnshaw 3. Hindley Earnshaw 4. Edgar Linton 5. Nelly Dean 6. Lockwood

Table 16: Central Characters Ordering Techniques

Relationships

Book	***Order***
Pride and Prejudice	1. Elizabeth & Darcy – Personal 2. Elizabeth & Jane – Family 3. Elizabeth & Lydia – Family 4. Elizabeth & Mrs. Bennet – Family 5. Jane & Mr. Bingley – Personal 6. Mr. Bingley & Mr. Darcy - Personal
Alice in Wonderland	1. Alice & The White Rabbit - Personal 2. Alice & Queen of Hearts - Personal 3. Alice & Cheshire Cat - Personal 4. Alice & Mad Hatter - Personal
Emma	1. Emma Woodhouse & Mr. Knightley - Personal 2. Emma Woodhouse & Harriet Smith - Personal 3. Emma Woodhouse & Mr. Woodhouse - Family 4. Frank Churchill & Jane Fairfax - Personal 5. Emma Woodhouse & Jane Fairfax - Personal 6. Emma Woodhouse & Frank Churchill - Personal 7. Emma Woodhouse & Rev. Elton - Other 8. Mr. Knightley & Mr. Woodhouse - Personal
Great Expectations	1. Phillip 'Pip' Pirrip & Estella - Personal 2. Phillip 'Pip' Pirrip & Joseph 'Joe' Gargery - Family 3. Phillip 'Pip' Pirrip & Abel Magwitch - Other 4. Estella & Miss Havisham - Family 5. Estella & Abel Magwitch – Family
Jane Eyre	1. Jane Eyre & Mr. Rochester - Personal 2. Jane Eyre & Mrs. Reed - Family 3. Jane Eyre & Mr. Rochester – Business 4. Mr. Rochester & Bertha - Family 5. Jane Eyre & St. John Rivers - Family
Little Women	1. Jo & Beth - Family

	2. Jo & Meg – Family 3. Jo & Amy - Family 4. Jo & Laurie - Personal
Peter Pan	(1) Peter Pan & Captain Hook - Personal (1) Peter Pan & Captain Hook - Other (2) Peter Pan & Wendy - Personal (3) Wendy & Michael – Family (3) Peter Pan & Tinkerbell - Personal
Wuthering Heights	1. Cathy & Heathcliff - Personal 2. Cathy & Edgar Linton - Family 3. Cathy & Hindley Earnshaw - Family 4. Heathcliff & Edgar Linton - Personal

Table 17: Central Characters Ordering Techniques

Appendix C: Collected Data

Data Collection 1st Attempt

Data Collection 2nd Attempt

OM 1 – Ordering Method 1 Rank

OM 2 – Ordering Method 2 Rank

Statistically Irrelevant Data (excluded from final set)

Central Characters

Pride and Prejudice

																																						OM 1	OM 2
Read Book	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			
Saw Film			Y		Y	Y		Y	Y	Y	Y	Y	Y		Y	Y		Y		Y	Y	Y						Y	Y	Y	Y	Y		Y	Y		Y		
Miss Elizabeth Bennet	2	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1.06
Mr. Darcy	1	2	1	2	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	1.94
Mr. Collins		6			7	6		4	7						7	7		6			7					7	5		6				7					100	10
Mr. Bingley	6	4		3	4	4	4	5			4	5		4	4	4	3	3	4	4	4	6	4	3	3	5	3	4	4	5	4		3	5	4			4	4.65
Miss Bingley		5			2												7			7											7								
Miss Jane Bennet	5	3	4			3	3	6		3	3	3	3	3	3	3	4	4	3	3	3	3	3	4		4	4	3	3	4	3		4	3	3		3	3	3.83
Lady Catherine de Bourgh									4			6							7					7										7					
Mr. Wickham	7			7	6	7	7		5	7	7				6	5		5		5		5	5	5		3	6		5	3	6		5	4	5		4	6	7.53
Lydia Bennet			6		5		6	7		6							6		6	6		4	6			6	7		7	6	5		6	6	6			7	11.86
Mrs. Bennet	3		5	6			5		3	4	5				5		5	7			5	7	7	6						7							6	100	11.54
Mr. Bennet	4		3	5		5		3		5	6	4							5		6														7		5	100	14.21
Kitty Bennet																																					7		
Charlotte Lucas									6			7				6																							

Peter Pan

					OM 1	OM 2
Read Book		Y	Y	Y		
Saw Film	Y	Y	Y	Y		
Peter Pan	1	1	1	1	1	1
Wendy (Darling)	2		2	2	2	2.66
Captain Hook	3	2	3	3	3	2.75
Mr. Darling	4					
Tinkerbell	7	3	4	4	4	4
The Lost Boys			4	5		
John (Darling)	5		5		52.5	10
Michael (Darling)	6		5	6	6	7.54
Crocodile			6			

Jane Eyre

							OM 1	OM 2
Read Book	Y	Y	Y	Y	Y			
Saw Film	Y	Y						
Jane Eyre	1	1	1	1	1		1	1
Mr. Rochester	2	2	2	2	2		2	2
Bertha (Rochester's wife)		3	5	5	3		5	5
Helen Burns				3			100	15
St. John Rivers	4			4			100	10
Blanche Ingram			4	6			100	12.5
Bessie				7				
Adelle			6					
Mrs. Reed	3		7				100	12.5

Great Expectations

							OM 1	OM 2
Read Book		Y	Y	Y	Y	Y		
Saw Film		Y	Y					
Pip	1	1	1	1	1	1	1	1
Estella	2	2	2	5	2	2	2	2.5
Miss Havisham	3	3	3	4	3	4	3	3.33
Magwich / Abel Magwitch	4	5	5	3		3	4	4.81
Joe (Gargery)	5	4	4	2	4	5	4.5	4
Mr. Jaggers	6		7	7			53.5	13.32
Biddy	7	6				7	53.5	13.32
Herbert		7		6		6	53.5	12.66
Mrs. Joe			6					

Wuthering Heights

						OM 1	OM 2
Read Book	Y	Y		Y	Y		
Saw Film	Y			Y			
Heathcliff	2	1	1	1	1	1	1
Catherine Earnshaw / Cathy	1	2	2	2	2	2	2
Hareton Earnshaw (Hindley's son)					3		
Hindley Earnshaw		4	3	6	4	3	4.25
Linton					5		
Edgar Linton		3		4		52	7
Isabella Linton					7		
Catherine Linton				7			
Nelly Dean		5		3		52.5	8
Lockwood		6		5		53	11
Mr. Earnshaw (Hindley's father)					6		

Alice in Wonderland

					OM 1	OM 2
Read Book	Y	Y	Y	Y		
Saw Film	Y		Y	Y		
Alice	1	1	1	1	1	1
Rabbit	2	2	2	2	2	2
Queen	3		3	3	3	4
Mad hatter	4	4	4		4	5.33
Dormouse	5					
Humpty Dumpty		3				
Cheshire Cat		5	5	4	5	6.21
March Hare		6				
Caterpillar		7				
Tweedle Dee			6			
Tweedle Dum			7			

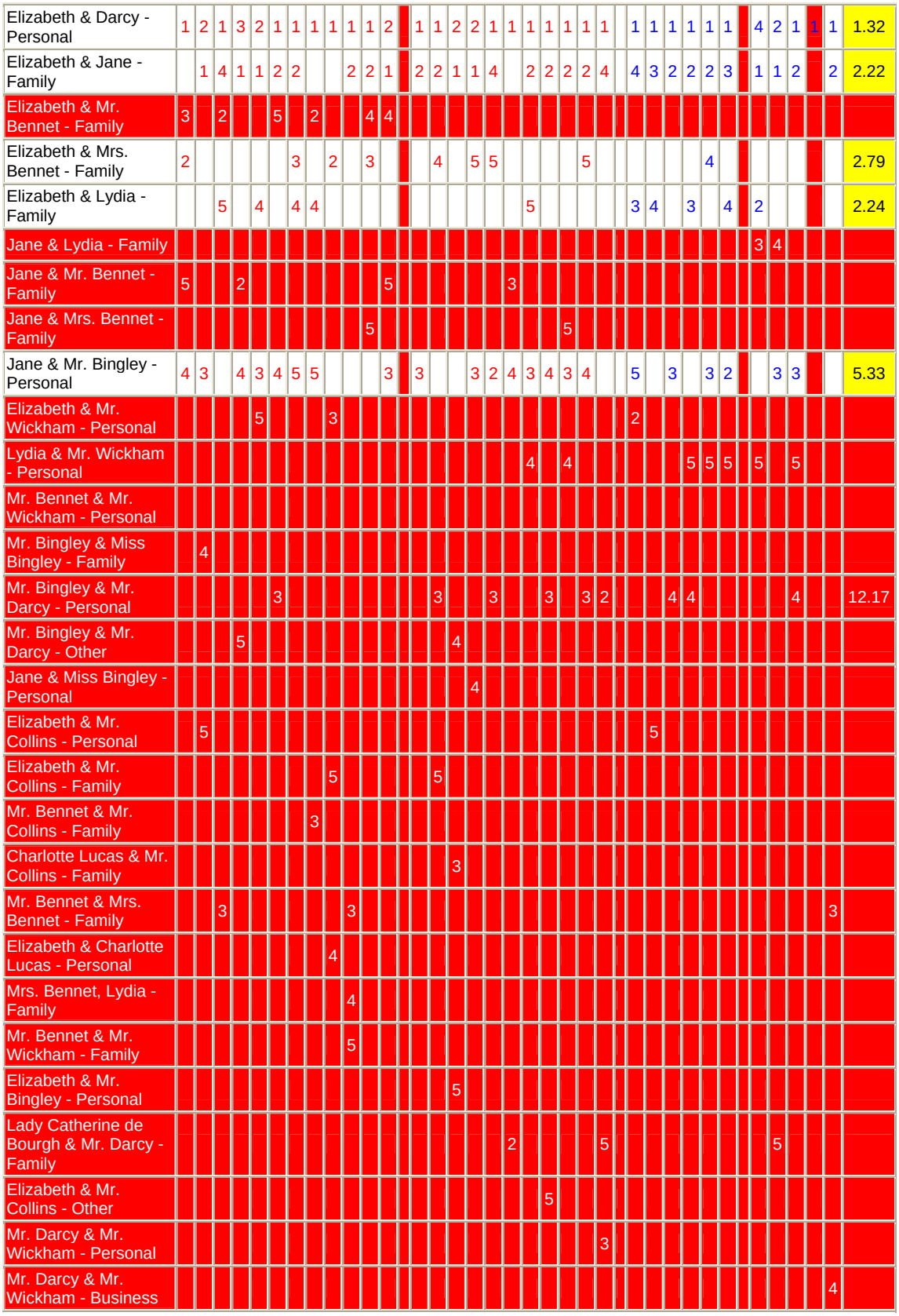
Emma

								OM 1	OM 2
Read Book	Y	Y	Y	Y	Y	Y	Y		
Saw Film	Y		Y	Y					
Emma Woodhouse	1	1	1	1	1	1	1	1	1
Mr. (George) Knightley	2		2	2	2	2	2	2	2
Harriet Smith	4		3	3	4	4	3	3.5	3.5
Jane Fairfax				4	5	3	4	4.5	5.97
Frank Churchill			4	5	3	6	5	5	5.54
Rev. Elton			6	6		5		53	11.32
Mrs. Weston, Miss Taylor						7			
Mr. Woodhouse	3		7		6			53.5	10.66
Robert Martin					7				
Miss Bates			5	7				100	18.18

Little Women

						OM 1	OM 2
Read Book	Y	Y	Y	Y	Y		
Saw Film				Y			
Jo March	1	1	1	1	1	1	1
Meg March	4	3	4	4	2	4	3.4
Laurie	3	5		5	3	5	5
Beth March	5	2	2	2	4	2	3
Amy March	2	4	3	3	5	3	3.4
Hannah					7		
Friedrich Bhaer				6			
Mr. March	6	6		7	6	6	7.81
John Brooke	7						

Relationships
Pride and Prejudice



Peter Pan

					OM 2
Wendy & Michael - Family			3	1	4
Peter Pan & Captain Hook - Personal			1	2	3
Peter Pan & Tinkerbell - Personal		2	4	3	4
Peter Pan & Wendy - Personal	1		2	4	3.11
Captain Hook & The Lost Boys - Personal				5	
Wendy & John - Family			3		
Peter Pan & Tinkerbell - Other	5				
Peter Pan & Crocodile - Personal			5		
Peter Pan & Captain Hook - Other	2	1			3
Wendy & Captain Hook - Other	3				
Peter Pan & Mr. Darling – Family ***	4				

Jane Eyre

							OM 2
Jane Eyre & Mr. Rochester - Business			1		1		2.5
Mr. Rochester & Bertha - Family		2	4	4	2		3.75
Jane Eyre & Mr. Rochester - Family					3		
Jane Eyre & Mr. Rochester - Personal	1	1		1			1.66
Jane Eyre & St. John Rivers - Family	3			2			6.25
Jane Eyre & Helen Burns - Personal				3			
Jane Eyre & Mrs. Reed - Family	2		2				2
Mr. Rochester & Blanche Ingram - Personal			3				

Great Expectations

							OM 2
Phillip 'Pip' Pirrip & Estella - Personal	1		1	4	2	1	2.16
Phillip 'Pip' Pirrip & Abel Magwitch - Other			4	2	1	2	3.37
Estella & Abel Magwitch - Family		4				3	10.5
Phillip 'Pip' Pirrip & Joseph 'Joe' Gargery - Family	5	1	2	1	4	4	2.83
Estella & Miss Havisham - Family	4			3		5	8
Estella & Miss Havisham - Other					3		
Abel Magwitch & Mr. Jaggers - Business				4			
Estella & Miss Havisham - Personal			3				
Phillip 'Pip' Pirrip & Mr. Jaggers - Business	2		5				
Joseph 'Joe' Gargery & Biddy - Personal		2					
Phillip 'Pip' Pirrip & Herbert Pocket - Personal		3					
Phillip 'Pip' Pirrip & Abel Magwitch - Business		5					
Phillip 'Pip' Pirrip & Miss Havisham - Personal	3						

Wuthering Heights

						OM 2
Cathy & Heathcliff - Personal	1	1	1	1		1.33
Heathcliff & Edgar Linton - Personal		2		5		7
Cathy & Edgar Linton - Family		3		2	1	2.66
Cathy & Hindley Earnshaw - Family		4	2		2	3.55
Hindley Earnshaw & Nelly Dean - other		5				
Cathy & Nelly Dean - Personal				3		
Heathcliff & Hindley Earnshaw - Family				4		
Isabella Linton & Hindley Earnshaw - Family					3	
Heathcliff & Cathy's father - Family					4	
Heathcliff & Isabella Linton - Family					5	

Alice in Wonderland

					OM 2
Alice & The White Rabbit - Personal	1		5	1	2.33
Alice & Queen of Hearts - Personal	2			2	3
Alice & Cheshire Cat - Personal			2	3	3.75
Tweedledee & Tweedledum - Family			1		
Alice & Mad Hatter - Personal	3		3		4.5

Emma

								OM 2
Emma Woodhouse & Mr. Knightley - Personal	1		3	1	1	1	1	1.33
Mr. Knightley & Harriet Smith - Personal							2	
Emma Woodhouse & Harriet Smith - Personal	3		2	2		3	3	3.12
Emma Woodhouse & Jane Fairfax - Personal						2	4	9
Emma Woodhouse & Frank Churchill - Personal			5			5	5	10
Emma Woodhouse & Rev. Elton - Personal	5							
Frank Churchill & Jane Fairfax - Personal				3	2			7.5
Harriet Smith & Robert Martin - Personal					3			
Emma Woodhouse & Mr. Woodhouse - Family	2		1		4			4.66
Mr. Knightley & Mr. Woodhouse - Personal	4				5			13.5
Emma Woodhouse & Rev. Elton - Other				4		4		12
Emma Woodhouse & Miss Bates - Other				5				
Emma Woodhouse & Miss Bates - Personal			4					

Little Women

Jo & Amy - Family	1				3	4
Jo & Meg - Family	2	2			1	2.66
Jo & Beth - Family	3	1		1	2	1.75
Jo & Laurie - Personal	4	4		5	5	4.5
Amy & Laurie - Personal	5					
Beth & Meg - Family		3				
Beth & Amy - Family				2		
Amy & Meg - Family				3		
Amy & Laurie - Family				4		
Jo & Mrs. March - Family					4	

Appendix D: Results

Central Characters

NER Results

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Pride and Prejudice	22.2%	66.6%	100%	100%
Peter Pan	0%	0%	0%	0%
Jane Eyre	0%	0%	0%	0%
Average	**7.4%**	**22.2%**	**33.3%**	**33.3%**
Average excluding books written in the first person	**11.1%**	**33.3%**	**50%**	**50%**

Table 18: NER absolute order results on the development set

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Wuthering Heights	0%	0%	0%	0%
Great Expectations	12.5%	0%	0%	0%
Little Women	33.3%	33.3%	50%	100%
Emma	25%	66.6%	50%	100%
Alice in Wonderland	0%	0%	0%	0%
Average	**14.1%**	**19.9%**	**20%**	**40%**
Average excluding books written in the first person	**19.4%**	**33.3%**	**33.3%**	**66.6%**
Average excluding books with non-human characters	**17.7%**	**24.9%**	**25%**	**50%**
Average Excluding all exceptions	**29.1%**	**49.9%**	**50%**	**100%**

Table 19: NER absolute order results on the test set

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Pride and Prejudice	77.7%	100%	100%	100%
Peter Pan	66.6%	66.6%	100%	100%
Jane Eyre	42.8%	66.6%	100%	100%
Average	**62.3%**	**77.7%**	**100%**	**100%**
Average excluding books written in the first person	**72.1%**	**83.3%**	**100%**	**100%**

Table 20: NER coverage results on the development set

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Wuthering Heights	66.6%	100%	100%	100%
Great Expectations	75%	66.6%	100%	100%
Little Women	83.3%	100%	100%	100%
Emma	62.5%	100%	100%	100%
Alice in Wonderland	0%	0%	0%	0%
Average	**57.4%**	**73.3%**	**40%**	**80%**
Average excluding books written in the first person	**48.6%**	**66.6%**	**20%**	**66.6%**
Average excluding books with non-human characters	**71.8%**	**91.6%**	**100%**	**100%**
Average excluding all exceptions	**72.9%**	**100%**	**100%**	**100%**

Table 21: NER coverage results on the test set

Book	***Precision***	***Recall***	***F-score***
Pride and Prejudice	80%	88%	83%
Peter Pan	40%	66%	49%
Jane Eyre	30%	42%	35%
Average	**50%**	**65.3%**	**55.6%**
Average excluding books written in the first person	**60%**	**77%**	**66%**

Table 22: NER F-score results on the development set

Book	***Precision***	***Recall***	***F-score***
Wuthering Heights	40%	66%	49%
Great Expectations	60%	75%	66%
Little Women	50%	83%	62%
Emma	50%	62%	55%
Alice in Wonderland	0%	0%	---
Average	**40%**	**57.2%**	**46.4%**
Average excluding books written in the first person)	**33.3%**	**48.3%**	**39%**
Average excluding books with non-human characters	**50%**	**71.5%**	**58%**
Average excluding all exceptions	**50%**	**72.5%**	**58.5%**

Table 23: NER F-score results on the test set

tf-idf Results

Using co-reference by tf

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Pride and Prejudice	55%	100%	100%	100%
Peter Pan	50%	100%	100%	100%
Jane Eyre	33%	66%	100%	100%
Average	**46.6%**	**88.6%**	**100%**	**100%**
Average excluding books written in the first person	**52.5%**	**100%**	**100%**	**100%**

Table 24: tf-idf absolute order results on the development set

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Wuthering Heights	0%	0%	0%	0%
Great Expectations	0%	0%	0%	0%
Little Women	20%	33.3%	50%	100%
Emma	12.5%	33.3%	50%	100%
Alice in Wonderland	20%	33.3%	50%	100%
Average	**10.5%**	**19.9%**	**30%**	**60%**
Average excluding books written in the first person	**17.5%**	**33.3%**	**50%**	**100%**
Average excluding books with non-human characters	**8.1%**	**16.6%**	**25%**	**50%**
Average excluding all exceptions	**16.2%**	**33.3%**	**50%**	**100%**

Table 25: tf-idf absolute order results on the test set

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Pride and Prejudice	88.8%	100%	100%	100%
Peter Pan	100%	100%	100%	100%
Jane Eyre	42.8%	66.6%	100%	100%
Average	**77.2%**	**88.8%**	**100%**	**100%**
Average excluding books written in the first person	**94.4%**	**100%**	**100%**	**100%**

Table 26: tf-idf coverage results on the development set

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Wuthering Heights	66.5%	100%	100%	100%
Great Expectations	87.5%	100%	100%	100%
Little Women	80%	100%	100%	100%
Emma	75%	100%	100%	100%
Alice in Wonderland	80%	100%	100%	100%
Average	**77.8%**	**100%**	**100%**	**100%**
Average excluding books written in the first person	**78.3%**	**100%**	**100%**	**100%**
Average excluding books with non-human characters	**77.2%**	**100%**	**100%**	**100%**
Average excluding all exceptions	**77.5%**	**100%**	**100%**	**100%**

Table 27: tf-idf coverage results on the test set

Book	***Precision***	***Recall***	***F-score***
Pride and Prejudice	80%	88%	83%
Peter Pan	60%	100%	75%
Jane Eyre	30%	42%	35%
Average	**56.6%**	**76.6%**	**64.3%**
Average (excluding books written in the first person)	**70%**	**94%**	**79%**

Table 28: tf-idf F-score results on the development set

Book	***Precision***	***Recall***	***F-score***
Wuthering Heights	40%	66%	49%
Great Expectations	70%	87%	77%
Little Women	40%	80%	53%
Emma	60%	75%	66%
Alice in Wonderland	40%	80%	53%
Average	**50%**	**77.6%**	**59.6%**
Average excluding books written in the first person	**46.6%**	**78.3%**	**57.3%**
Average excluding books with non-human characters	**52.5%**	**77%**	**61.2%**
Average excluding all exceptions	**50%**	**77.5%**	**59.5%**

Table 29: tf-idf F-score results on the test set

Using co-reference by contextual information

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Pride and Prejudice	44.4%	66.6%	100%	100%
Peter Pan	66.6%	100%	100%	100%
Jane Eyre	0%	0%	0%	0%
Average	**37%**	**55.5%**	**66.6%**	**66.6%**
Average excluding books written in the first person	**55.5%**	**83.3%**	**100%**	**100%**

Table 30: tf-idf absolute order results on the development set

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Wuthering Heights	16.6%	33.3%	50%	100%
Great Expectations	28.5%	66.6%	100%	100%
Little Women	40%	66.6%	50%	100%
Emma	12.5%	33.3%	50%	100%
Alice in Wonderland	20%	33.3%	50%	100%
Average	**23.5%**	**46.6%**	**60%**	**100%**
Average excluding books written in the first person	**24.1%**	**44.4%**	**50%**	**100%**
Average excluding books with non-human characters	**24.4%**	**49.9%**	**62.5%**	**100%**
Average excluding all exceptions	**26.2%**	**49.9%**	**50%**	**100%**

Table 31: tf-idf absolute order results on the test set

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Pride and Prejudice	88.8%	100%	100%	100%
Peter Pan	83.3%	100%	100%	100%
Jane Eyre	57.1%	66.6%	100%	100%
Average	**76.4%**	**88.8%**	**100%**	**100%**
Average excluding books written in the first person	**86%**	**100%**	**100%**	**100%**

Table 32: tf-idf coverage results on the development set

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Wuthering Heights	66.6%	100%	100%	100%
Great Expectations	87.5%	100%	100%	100%
Little Women	66.6%	66.6%	50%	100%
Emma	75%	100%	100%	100%
Alice in Wonderland	80%	100%	100%	100%
Average	**75.1%**	**93.3%**	**90%**	**100%**
Average excluding books written in the first person	**73.8%**	**88.8%**	**83.3%**	**100%**
Average excluding books with non-human characters	**73.9%**	**91.6%**	**87.5%**	**100%**
Average excluding all exceptions	**70.8%**	**83.3%**	**75%**	**100%**

Table 33: tf-idf coverage results on the test set

Book	***Precision***	***Recall***	***F-score***
Pride and Prejudice	80%	88%	83%
Peter Pan	40%	66%	49%
Jane Eyre	40%	57%	47%
Average	**53.3%**	**70.3%**	**59.6%**
Average excluding books written in the first person	**60%**	**77%**	**66%**

Table 34: tf-idf F-score results on the development set

Book	***Precision***	***Recall***	***F-score***
Wuthering Heights	40%	66%	49%
Great Expectations	70%	87%	77%
Little Women	40%	66%	49%
Emma	50%	62%	55%
Alice in Wonderland	40%	80%	53%
Average	**48%**	**72.2%**	**56.6%**
Average excluding books written in the first person	**43.3%**	**69.3%**	**52.3%**
Average excluding books with non-human characters	**50%**	**70.2%**	**57.5%**
Excluding all exceptions	**45%**	**64%**	**52%**

Table 35: tf-idf F-score results on the test set

Frequency Counts Results

Book	*Superset*	*Top 3*	*Top 2*	*Top 1*
Pride and Prejudice	66.6%	100%	100%	100%
Peter Pan	50%	100%	100%	100%
Jane Eyre	0%	0%	0%	0%
Average	**38.8%**	**66.6%**	**66.6%**	**66.6%**
Average excluding books written in the first person	**58.3%**	**100%**	**100%**	**100%**

Table 36: Frequency count absolute order results on the development set

Book	*Superset*	*Top 3*	*Top 2*	*Top 1*
Wuthering Heights	16.6%	33.3%	50%	100%
Great Expectations	14.28%	0%	0%	0%
Little Women	20%	33.3%	50%	100%
Emma	12.5%	33.3%	50%	100%
Alice in Wonderland	20%	33.3%	50%	100%
Average	**16.6%**	**26.6%**	**40%**	**80%**
Average excluding books written in the first person	**17.5%**	**33.3%**	**50%**	**100%**
Average excluding books with non-human characters	**15.8%**	**24.9%**	**37.5%**	**75%**
Average excluding all exceptions	**16.2%**	**33.3%**	**50%**	**100%**

Table 37: Frequency count absolute order results on the test set

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Pride and Prejudice	88.8%	100%	100%	100%
Peter Pan	100%	100%	100%	100%
Jane Eyre	57.1%	66.6%	100%	100%
Average	**81.9%**	**88.8%**	**100%**	**100%**
Average excluding books written in the first person	**94.4%**	**200%**	**100%**	**100%**

Table 38: Frequency count coverage results on the development set

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Wuthering Heights	66.6%	100%	100%	100%
Great Expectations	87.5%	100%	100%	100%
Little Women	83.3%	100%	100%	100%
Emma	75%	100%	100%	100%
Alice in Wonderland	80%	100%	100%	100%
Average	**78.4%**	**100%**	**100%**	**100%**
Average excluding books written in the first person	**79.4%**	**100%**	**100%**	**100%**
Average excluding books with non-human characters	**78.1%**	**100%**	**100%**	**100%**
Average excluding all exceptions	**79.1%**	**100%**	**100%**	**100%**

Table 39: Frequency count coverage results on the test set

Book	***Precision***	***Recall***	***F-score***
Pride and Prejudice	80%	88%	83%
Peter Pan	60%	100%	75%
Jane Eyre	40%	66%	49%
Average	**60%**	**84.6%**	**69%**
Average excluding books written in the first person	**70%**	**94%**	**79%**

Table 40: Frequency count F-score results on the development set

Book	***Precision***	***Recall***	***F-score***
Wuthering Heights	40%	66%	49%
Great Expectations	70%	87%	77%
Little Women	50%	83%	62%
Emma	60%	75%	66%
Alice in Wonderland	40%	80%	53%
Average	**52%**	**78.2%**	**61.4%**
Average excluding books written in the first person	**50%**	**79.3%**	**60.3%**
Average excluding books with non-human characters	**55%**	**77.7%**	**63.5%**
Average excluding all exceptions	**55%**	**79%**	**64%**

Table 41: Frequency count F-score results on the test set

Tested Set	*Baseline*	*Tf-idf (1)*[33]	*tf-idf (2)*[34]	*Frequency*
Set	14.10%	10.50%	23.50%	16.60%
Set excluding books written in the first person	19.40%	17.50%	24.10%	17.50%
Set excluding books with non-human characters	17.70%	8.10%	24.40%	15.80%
Set Excluding exceptions	29.10%	16.20%	26.20%	16.20%
Top 3	19.90%	19.90%	46.60%	26.60%
Top 3 excluding books written in the first person	33.30%	33.30%	44.40%	33.30%
Top 3 excluding books with non-human characters	24.90%	16.60%	49.90%	24.90%
Top 3 excluding all exceptions	49.90%	33.30%	49.90%	33.30%
Top 2	20.00%	30.00%	60.00%	40.00%
Top 2 excluding books written in the first person	33.30%	50.00%	50.00%	50.00%
Top 2 excluding books with non-human characters	25.00%	25.00%	62.50%	37.50%
Top 2 excluding all exceptions	50.00%	50.00%	50.00%	50.00%
Top 1	40.00%	60.00%	100.00%	80.00%
Top 1 excluding books written in the first person	66.60%	100.00%	100.00%	100.00%
Top 1 excluding books with non-human characters	50.00%	50.00%	100.00%	75.00%
Top 1 excluding all exceptions	100.00%	100.00%	100.00%	100.00%

Table 42: Overall absolute order results on the test set

[33] tf-idf with co-reference resolution by contextual information.

[34] tf-idf with co-reference resolution by tf scores.

	Baseline	***tf-idf (1)***[35]	***tf-idf (2)***[36]	***Frequency***
Set	57.40%	77.80%	75.10%	78.40%
Set excluding books written in the first person	48.60%	78.30%	73.80%	79.40%
Set excluding books with non-human characters	71.80%	77.20%	73.90%	78.10%
Set Excluding exceptions	72.90%	77.50%	70.80%	79.10%
Top 3	73.30%	100.00%	93.30%	100.00%
Top 3 excluding books written in the first person	66.60%	100.00%	88.80%	100.00%
Top 3 excluding books with non-human characters	91.60%	100.00%	91.60%	100.00%
Top 3 excluding all exceptions	100.00%	100.00%	83.30%	100.00%
Top 2	40.00%	100.00%	90.00%	100.00%
Top 2 excluding books written in the first person	20.00%	100.00%	83.30%	100.00%
Top 2 excluding books with non-human characters	100.00%	100.00%	87.50%	100.00%
Top 2 excluding all exceptions	100.00%	100.00%	75.00%	100.00%
Top 1	80.00%	100.00%	100.00%	100.00%
Top 1 excluding books written in the first person	66.60%	100.00%	100.00%	100.00%
Top 1 excluding books with non-human characters	100.00%	100.00%	100.00%	100.00%
Top 1 excluding all exceptions	100.00%	100.00%	100.00%	100.00%

Table 43: Overall coverage results on test set

[35] tf-idf with co-reference resolution by contextual information

[36] tf-idf with co-reference resolution by t scores

	Baseline	***tf-idf (1)***[37]	***tf-idf (2)***[38]	***Frequency***
Precision	40.00%	50.00%	48.00%	52.00%
Precision exc 1st person	33.30%	46.60%	43.30%	50.00%
Precision exc non human	50.00%	52.50%	50.00%	55.00%
Precision exc all	50.00%	50.00%	45.00%	55.00%
Recall	57.20%	77.60%	72.20%	78.20%
Recall exc 1st person	48.30%	78.30%	69.30%	79.30%
Recall exc non human	71.50%	77.00%	70.20%	77.70%
Recall exc all	72.50%	77.50%	64.00%	79.00%
F-score	46.40%	59.60%	56.60%	61.40%
F-score excluding 1st person	**39.00%**	**57.30%**	**52.30%**	**60.30%**
F-score excluding non human	**58.00%**	**61.20%**	**57.50%**	**63.50%**
F-score excluding all	**58.50%**	**59.50%**	**52.00%**	**64.00%**

Table 44: F-score results on test set

[37] tf-idf with co-reference resolution by contextual information

[38] tf-idf with co-reference resolution by tf scores

Agreement between tf-idf and frequency

Absolute Order

Book	Superset	Top 3	Top 2	Top 1
Pride and Prejudice	100%	100%	100%	100%
Peter Pan	10%	33.3%	50%	100%
Jane Eyre	30%	33.3%	50%	100%
Wuthering Heights	60%	100%	100%	100%
Great Expectations	20%	0%	0%	0%
Little Women	10%	33.3%	50%	100%
Emma	60%	100%	100%	100%
Alice in Wonderland	10%	33.3%	50%	100%
Average	**37.5%**	**54.1%**	**62.5%**	**87.5%**

Table 45: Absolute order agreement between tf-idf and frequency counts

Coverage

Book	***Superset***	***Top 3***	***Top 2***	***Top 1***
Pride and Prejudice	100%	100%	100%	100%
Peter Pan	90%	100%	100%	100%
Jane Eyre	80%	66.6%	100%	100%
Wuthering Heights	100%	100%	100%	100%
Great Expectations	100%	100%	100%	100%
Little Women	80%	100%	100%	100%
Emma	90%	100%	100%	100%
Alice in Wonderland	90%	100%	100%	100%
Average	**91.2%**	**95.8%**	**100%**	**100%**

Table 46: Coverage agreement between tf-idf and frequency counts

Relationships

Relationships are evaluated in terms of coverage and all the tables in the section contain coverage results.

Coverage

Book	*Superset*	*Top 3*	*Top 2*	*Top 1*
Pride and Prejudice	33.3%	66.6%	100%	100%
Peter Pan	66.6%	66.6%	100%	100%
Jane Eyre	0%	0%	0%	0%
Average	**33.3%**	**44.4%**	**66.6%**	**66.6%**
Average excluding books written in the first person	**49.9%**	**66.6%**	**100%**	**100%**

Table 47: Relationships coverage results for the development set

Book	*Superset*	*Top 3*	*Top 2*	*Top 1*
Wuthering Heights	75%	66.6%	100%	100%
Great Expectations	60%	33.3%	50%	0%
Little Women	100%	100%	100%	100%
Emma	50%	100%	100%	100%
Alice in Wonderland	50%	66.6%	100%	100%
Average	**67%**	**73.3%**	**90%**	**80%**
Average excluding books written in the first person	**66.6%**	**88.8%**	**100%**	**100%**
Average excluding books with non-human characters	**71.2%**	**74.9%**	**87.5%**	**75%**
Excluding all exceptions	**75%**	**100%**	**100%**	**100%**

Table 48: Relationships coverage results for the test set

Random Selection

Book	*Precision*	*Recall*	*F-score*
Wuthering Heights	10%	25%	14.2%
Great Expectations	20%	40%	26.6%
Little Women	20%	50%	28.5%
Emma	20%	33.3%	24.9%
Alice in Wonderland	0%	0%	0%
Average	**14%**	**29.6%**	**18.8%**
Average excluding books written in the first person	**13.3%**	**27.7%**	**17.8%**
Average excluding books with non-human characters	**17.5%**	**37.7%**	**23.5%**
Excluding all exceptions	**20%**	**41.6%**	**26.7%**

Table 49: Relationships random selection results for the test set

Majority Default Selection

Book	*Precision*	*Recall*	*F-score*
Wuthering Heights	20%	50%	28.5%
Great Expectations	10%	20%	13.3%
Little Women	30%	75%	42.8%
Emma	40%	50%	44.4%
Alice in Wonderland	20%	50%	28.5%
Average	**24%**	**49%**	**31.5%**
Average excluding books written in the first person	**30%**	**58.3%**	**38.5%**
Average excluding books with non-human characters	**25%**	**48.7%**	**32.2%**
Excluding all exceptions	**35%**	**62.5%**	**43.6%**

Table 50: Relationships majority default selection results for the test set

Rule Based

Book	*Precision*	*Recall*	*F-score*
Wuthering Heights	0%	0%	0%
Great Expectations	10%	20%	13.3%
Little Women	30%	60%	40%
Emma	0%	0%	0%
Alice in Wonderland	0%	0%	0%
Average	**8%**	**16%**	**10.6%**
Average excluding books written in the first person	**10%**	**20%**	**13.3%**
Average excluding books with non-human characters	**10%**	**20%**	**13.3%**
Excluding all exceptions	**15%**	**30%**	**20%**

Table 51: Relationships rule based selection results for the test set

Summaries

Random Select

Book	*Precision*	*Recall*	*F-score*
Wuthering Heights	20%	50%	28.5%
Great Expectations	20%	40%	26.6%
Little Women	10%	25%	14.2%
Emma	33.3%	37.5%	34.8%
Alice in Wonderland	14.2%	25%	17.9%
Average	**19.5%**	**35.5%**	**24.4%**
Average excluding books written in the first person	**19.1%**	**29.1%**	**22.3%**
Average excluding books with non-human characters	**20.8%**	**38.1%**	**26%**
Excluding all exceptions	**21.6%**	**31.2%**	**24.5%**

Table 52: Summary results for random selection

Majority Default Selection

Book	***Precision***	***Recall***	***F-score***
Wuthering Heights	20%	50%	28.5%
Great Expectations	10%	20%	13.3%
Little Women	10%	25%	14.2%
Emma	50%	62.5%	55.3%
Alice in Wonderland	28.5%	50%	35.8%
Average	**23.7%**	**41.5%**	**29.4%**
Average excluding books written in the first person	**29.5%**	**45.8%**	**35.1%**
Average excluding books with non-human characters	**22.5%**	**39%**	**27.83%**
Excluding all exceptions	**30%**	**43.7%**	**34.7%**

Table 53: Summary results for majority default selection

Concept Rules

Book	***Precision***	***Recall***	***F-score***
Wuthering Heights	20%	50%	28.5%
Great Expectations	30%	60%	40%
Little Women	10%	25%	14.2%
Emma	0%	0%	0%
Alice in Wonderland	0%	0%	0%
Average	**12%**	**27%**	**16.5%**
Average excluding books written in the first person	**33.3%**	**8.3%**	**4.7%**
Average excluding books with non-human characters	**15%**	**33.7%**	**20.6%**
Excluding all exceptions	**5%**	**12.5%**	**7.1%**

Table 54: Summary results for rule based selection